ABSTRACT

Title of dissertation: INTEGRATED PRODUCTION-DISTRIBUTION SCHEDULING IN SUPPLY CHAINS

Guruprasad Pundoor, Doctor of Philosophy, 2005

Dissertation directed by: Dr. Zhi-Long Chen, Associate Professor
Robert H. Smith School of Business

We consider scheduling issues in different configurations of supply chains. The primary focus is to integrate production and distribution activities in the supply chain in order to optimize the tradeoff between total cost and service performance. The cost may be based on actual expenses such as the expense incurred during the distribution phase, and service performance can be expressed in terms of time based performance measures such as completion times and tardiness. Our goal is to achieve the following objectives: (i) To propose various integrated production-distribution scheduling models that closely mirror practical supply chain operations in some environments. (ii) To develop computationally effective optimization based solution algorithms to solve these models. (iii) To provide managerial insights into the potential benefits of coordination between production and distribution operations in a supply chain.

We analyze four different configurations of supply chains. In the first model, we consider a setup with multiple manufacturing plants owned by the same firm. The manufacturer receives a set of distinct orders from the retailers before a selling season, and needs to determine the order assignment, production schedule, and distribution schedule so as to optimize a certain performance measure of the supply chain. The second model deals with a supply chain consisting of one supplier and one or more customers, where the customers set due dates on the orders they place. The supplier has to come up with an integrated production-distribution schedule that optimizes the tradeoff between maximum tardiness and total distribution cost. In the third model, we study an integrated production and distribution scheduling model in a two-stage supply chain consisting of one or more suppliers, a warehouse, and a customer. The objective is to find jointly a cyclic production schedule at each supplier, a cyclic delivery schedule from each supplier to the warehouse, and a cyclic delivery schedule from the warehouse to the customer so that the customer demand for each product is satisfied fully at minimum total production, inventory and distribution cost. In the fourth model, we consider a system with one supplier and one customer with a set of orders placed at the beginning of the planning horizon. Unlike the earlier models, here each order can have a different size. Since the shipping capacity per batch is finite, we have to solve an integrated production-distribution scheduling and order-packing problem. Our objective is to minimize the number of delivery batches subject to certain service performance measures such as the average lead time or compliance with deadlines for the orders.

Keywords: supply chain, production and distribution scheduling, NP-completeness, linear and integer programming, heuristic, dynamic programming, worst-case analysis, asymptotic analysis, column generation, order packing.

INTEGRATED PRODUCTION-DISTRIBUTION SCHEDULING IN SUPPLY CHAINS

by

Guruprasad Pundoor

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Professor Zhi-Long Chen (Chairman/Advisor)
Professor Michael Ball
Professor Michael Fu
Professor Jeffrey Herrmann
Professor Itir Karaesmen

DEDICATION

To my parents

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Dr. Zhi-Long Chen, my advisor, whose guidance, enthusiastic support, and advice throughout this research project were invaluable. I was extremely lucky to have him as my advisor and I look forward to continue working with him in the future.

I am grateful to Dr. Michael Ball, Dr. Michael Fu, Dr. Jeffrey Herrmann (Dean's representative), and Dr. Itir Karaesmen for agreeing to serve on my committee; their suggestions at the proposal stage have helped hone my thesis.

I would like to thank the Robert H. Smith School of Business and the National Science Foundation (grants DMI-0196536 and DMI-0421637) for supporting me financially through my doctoral studies. In addition to this thesis, I have also had opportunities to work with other faculty members at the Smith School; these have further enriched my experience in graduate school. I would like to extend my special thanks to Dr. Arjang Assad, Dr. Bruce Golden, Dr. Itir Karaesmen and Dr. Raghavan. I am grateful to Dr. Anandalingam for the several rewarding discussions; his ever helpful and approachable nature has been a source of inspiration to me. I would like to thank Ms. Mary Slye at the Ph.D program office for her help with administrative matters.

Thanks are also due to my friends and colleagues who made my stay here for the last three years very memorable. Last but not least, very special thanks go to my parents who have helped me in all phases of my life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The primary difference between analyzing a supply chain and analyzing a production system or a distribution system is that in a supply chain, we may have to simultaneously consider different and sometimes conflicting objectives from different participants, or different departments within the same participant. For example, minimizing production costs at the production department may have to be carried out by taking into account the distribution costs at the distribution department. Similarly, minimizing distribution costs at the distribution department may have to consider the delivery lead time performance. Or, optimizing the distribution costs at a supplier by sending large shipments may have to put up with an increase in the inventory holding costs at the warehouse. Though production scheduling and distribution scheduling have separately been studied extensively, very little work has been done that integrates these two operations in supply chains. Supply chain level decision making is very crucial for most of the businesses that exist today. This opens up a very promising area of research.

In this work, we consider scheduling issues in different configurations of supply chains. The primary focus is to integrate production and distribution activities in the supply chain in order to optimize the tradeoff between total cost and service performance. The cost may be based on actual expenses such as the expense incurred during the distribution phase, and service performance can be expressed in terms of time based performance measures such as completion times and tardiness. Our goal is to achieve the following objectives: (i) To propose various integrated production-distribution scheduling models that closely mirror practical supply chain operations in some environments. (ii) To develop computationally effective optimization based solution algorithms to solve these models. Our solution approaches can be used as decision tools by practitioners. (iii) To provide managerial insights into the potential benefits of coordination between production and distribution operations in a supply chain. We make use of techniques ranging from simple first order conditions to mixed integer programming formulations. The different supply chain systems are discussed next.

1.1 Order Assignment and Scheduling in a Supply Chain with Multiple Suppliers Serving One Customer

Consider the global supply chain of a manufacturer with a number of manufacturing plants (suppliers). The manufacturer produces time-sensitive products, such as toys, fashion apparel, or high-tech products that typically have a large variety, a short life cycle, and are sold in a very short selling season. Because of high demand uncertainty of the products, retailers typically do not place orders until reliable market information is available shortly before a selling season. On the other hand, since there are significant markdowns for unsold products at the end of the selling season, the manufacturer runs a high risk if he/she starts production early before receiving orders from the retailers. As a result, the manufacturer would not start production until orders from the retailers have been placed shortly before the selling season. Due to the fact that there is only a limited amount of production time available, in order to deliver the orders to the retailers as soon as possible at a low cost, the manufacturer has to schedule the production and distribution operations in a coordinated and efficient manner. We consider a simplified version of the order assignment and scheduling problem faced by the manufacturer in the above-described supply chain. In this problem, the manufacturer receives a set of distinct orders from the retailers before a selling season, and needs to determine (i) which orders to be assigned to which plants, (ii) how to schedule the production of the assigned orders at each plant, and (iii) how to schedule the distribution of the completed orders from each plant to the distribution center (DC), so as to optimize a certain performance measure of the supply chain. Due to the variations in productivity and labor costs at different plants, the processing time and cost of an order are dependent on the plant to which it is assigned. Completed orders are delivered in shipments from the plants to the DC. Each shipment can carry up to a certain number of orders and is associated with a certain distribution time and a certain distribution cost. We consider the following four performance measures:

P1: Minimizing a weighted sum of the total lead time and total cost, i.e. $\alpha D_{\text{total}} + (1 - \alpha)TC$, where $\alpha \in [0, 1]$ is a given constant, representing the decision-maker's relative preference on D_{total} and TC.

P2: Minimizing the total cost TC subject to the constraint that the total lead time is no more than a given threshold, i.e. $D_{\text{total}} \leq D$, where D is a given constant.

P3: Minimizing a weighted sum of the maximum lead time and total cost, i.e. $\alpha D_{\max} + (1 - \alpha)TC$, where $\alpha \in [0, 1]$ is a given constant, as in problem P1.

P4: Minimizing the total cost TC subject to the constraint that the maximum lead time is no more than a given threshold, i.e. $D_{\max} \leq D$, where D is a given constant.

Here TC includes both the production and distribution costs, D_{total} represents the sum of lead times of all the orders, and $D_{\max}$ represents the maximum lead time among all the orders. We either prove that a problem is intractable, or provide an efficient exact algorithm for the problem. All the four problems are in general NP-hard, and fast heuristics have been proposed for each of them. Worst-case and asymptotic performance of two of the heuristics have been analyzed. Each heuristic has been evaluated computationally and the results show that each heuristic is in general capable of generating near optimal solutions. Some simplified polynomially solvable cases of the problem are also considered. We also compare the performance of the integrated approach by empirically testing our approach with an approach that optimizes the production and distribution parts independent of each other. It is not uncommon to find an improvement of 10% or more in the performance measure by choosing our integrated approach.

1.2 Optimizing the Tradeoff between Delivery Tardiness and Distribution Cost in a Supply Chain with One Supplier Serving Multiple Customers

The second model deals with a make-to-order production-distribution system with one supplier and one or more customers. The customers (e.g. distributors or retailers) often set due dates on the orders they place with the supplier and there is typically a penalty imposed on the supplier if the orders are not completed and delivered to the customers on time. Hence the supplier would like to meet the due dates as much as possible. Another factor the supplier has to consider is the total distribution cost for order delivery. Since different orders may have different due dates, delivering more orders on time might require the supplier to make a larger number of shipments leading to higher total distribution cost. Completed orders are delivered in batches to the customers. Since each customer is located at a distinct location, we assume that only orders from the same customer can be batched together to form a delivery shipment and orders from different customers must be delivered separately. The supplier has to find a production and distribution schedule that achieves some balance between delivery timeliness and total distribution cost.

We focus on the maximum tardiness as the measure for delivery timeliness. Hence the objective is to minimize the total cost, where total cost is given as a weighted sum of the total distribution cost and the maximum tardiness. The objective function is then defined as $\alpha T_{\max} + (1-\alpha)G,\ 0 \le \alpha \le 1$, where $T_{\max}$ is the maximum tardiness and G is the total distribution cost. G can be expressed as the sum of costs corresponding to each delivery batch. It can be seen that when α is close to 0, more emphasis is given to the total distribution cost and when α is close to 1, more emphasis is given to $T_{\max}$.

We study the solvability of various cases of the problem. We also analyze a special case where the processing times and the due dates are agreeable. Let p_{ij} and d_{ij} represent the processing time and due date of an order j from customer i respectively. In the case of agreeable processing times and due dates, if jobs u and v from customer i have processing times $p_{iu} \le p_{iv}$, then their due dates follow the relation $d_{iu} \le d_{iv}$. We give a polynomial time algorithm for the general problem with a single-customer. We also show that the multiple-customer problem for an arbitrary number is customers is NP-hard even when the processing times and the due dates are agreeable. We develop a fast and asymptotically optimal heuristic for the general case. We also evaluate the performance of the heuristic computationally by using lower bounds obtained by a column generation approach. Finally, we study the value of production-distribution integration by comparing our integrated approach with a sequential approach where scheduling decisions for order processing are made first, followed by order delivery decisions, without a joint consideration. Results show that the integrated approach leads to good improvements in performance under cases where the contribution due to the maximum tardiness is significant in the objective function value.

1.3 Joint Cyclic Production and Delivery Scheduling in a Two-Stage Supply Chain

In the third model, we study an integrated production and distribution scheduling model in a two-stage supply chain consisting of one or more suppliers, a warehouse, and a customer. The first two models looked at a make-to-order scenario over a finite horizon of time where each order is distinct. In this model, we consider an infinite horizon cyclic scenario where there is only one product and the demand rate at the customer is assumed to be constant over time. This model extends the concepts of economic lot sizing problems to jointly consider the delivery of product to the customer in a two-stage supply chain. The objective is to find jointly a cyclic production schedule at each supplier, a cyclic delivery schedule from each supplier to the warehouse, and a cyclic delivery schedule from the warehouse to the customer so that the customer demand for each product is satisfied fully at minimum total production, inventory and distribution cost. We study the problem under various production and delivery scheduling policies. One of the commonly made assumptions in the literature for this category of problems is that the cycle time at one stage is an integral multiple of the cycle time at its immediate successor. This assumption simplifies the inventory calculations and it can also be shown that the assumption is optimal under many cases. We also consider the special case where the production cycle time at the supplier is the same as the delivery cycle time from the supplier to the warehouse.

We give either optimal approaches or heuristic methods to solve the problem under two policies on production and delivery cycle times. Under policy (i), the production cycle time at each supplier is identical to the delivery cycle time from the supplier to the warehouse. Under policy (ii), the production cycle time at each supplier is an integer multiple of the delivery cycle time from that supplier to the warehouse, and the delivery cycle time from a supplier to the warehouse is an integer multiple of the delivery cycle time from the warehouse to the customer. For policy (i), we prove that there exists an optimal solution where the delivery cycle time from a supplier to the warehouse is an integer multiple of the delivery cycle time from the warehouse to the customer. Based on this property, we show that there is a closed-form optimal solution to the problem with a single supplier under policy (i), and develop an efficient heuristic for the problem with multiple suppliers. The problem under policy (ii) is solved by a heuristic approach. Both heuristics are shown to perform very well for an extensive set of test problems. We also computationally evaluate the value of warehouse in our two-stage supply chain.

An important use of this study is to make operational decisions regarding the delivery intervals in a two-stage supply chain. The models can also be used to make strategic decisions related to configuring or making changes to a supply chain. For example, we could use the heuristics to choose between a single-stage and a two-stage supply chain. Given that a warehouse has to be built, we could use this study to analyze the total costs corresponding to various locations of the potential warehouse. We could use the heuristics to analyze the trade-offs involved in moving an existing warehouse to a new location. This model can also be used to analyze the effect of reducing the setup cost or setup time on the performance of the entire supply chain.

1.4 Integrating Order Scheduling with Packing and Delivery in a One Supplier - One Customer Supply Chain

In the fourth model, we study integrated production-distribution scheduling in a make-to-order supply chain that consists of one supplier and one customer, where different orders may have different delivery capacity requirements. The supplier receives a set of orders from the customer at the beginning of the planning horizon. The supplier needs to process all the orders at a single production line, pack the completed orders to form delivery batches, and deliver the batches to the customer. Each order has a weight and the total weight of the orders that are packed in each delivery batch must not exceed a capacity limit. Each delivery batch incurs a fixed distribution cost regardless of the total weight it carries. The problem is to find jointly a schedule for order processing at the supplier, a way of packing completed orders to form delivery batches, and a delivery schedule from the supplier to the customer such that the total distribution cost is minimized subject to the constraint that a given customer service level is guaranteed. We consider two customer service constraints - meeting the given deadlines of the orders; or requiring the average delivery lead time of the orders to be within a given threshold. We consider the following different scenarios:

(i) Non-splittable production and delivery: An order cannot be split in terms of production or delivery, i.e. it is not allowed to preempt the processing of an order and a finished order must be delivered in one batch.

(ii) Non-splittable production, but splittable delivery: An order cannot be split in terms of production, but can be split in terms of delivery, i.e. no processing preemption is allowed, but a finished order can be split into multiple parts delivered in multiple batches.

(iii) Splittable production and delivery: An order can be split in terms of both production and delivery, i.e. both processing preemption and delivery split of an order are allowed.

We clarify the complexity of each problem by either proving its intractability or providing an efficient algorithm for it. We then develop fast heuristics for the intractable problems and analyze their worst-case performance. We propose column generation based approaches for finding lower bounds of the objective values of various problems, and use those bounds to evaluate the performance of the heuristics computationally. Our results indicate that all the heuristics are capable of generating near optimal solution quickly for the respective problems. We also consider two extensions: one in which inventory costs at the supplier are considered and another in which a fraction of the orders may dynamically arrive after the production has begun for the other orders. In the second case, it may be necessary to update an existing schedule in order to accommodate the new arrivals.

1.5 Literature Review

There is a huge body of literature on the production-distribution problems, models, networks, or systems. As pointed out in the survey by Chen (2004), many existing models study strategic or tactical levels of decisions, and very few have addressed integrated decisions at the detailed scheduling level. See Vidal and Goetschalckx (1997) and Owen and Daskin (1998) for reviews, and Jayaraman and Pirkul (2001), Dasci and Verter (2001), and Shen et al. (2003) for recent results in this area. A major portion of the broad literature in the production-distribution area is on the following two classes of problems: (i) Problems that integrate inventory replenishment decisions across multiple stages of the supply chain. See, among others, Williams (1983), Muckstadt and Roundy (1993), Pyke and Cohen (1994), Bramel et al. (2000), and Boyaci and Gallego (2001). (ii) Problems that integrate inventory and distribution decisions. See, among others, Burns et al. (1985), Speranza and Ukovich (1994), Chan et al. (1997), Bertazzi and Speranza (1999). These problems either ignore or oversimplify production operations (e.g. assuming instantaneous production without production time or capacity consideration). On the other hand, in our models, we deal at the operational level as opposed to the strategic or tactical levels and explicitly consider scheduling operations.

Our models are different from many existing models that integrate production and distribution decisions (e.g. Cohen and Lee 1988, Chandra and Fisher 1994, Hahm and Yano 1995, Fumero and Vercellis 1999, Sarmiento and Nagi 1999). In many existing models, inventory costs are a significant portion of the total cost, and production and distribution are indirectly linked through inventory and their linkage is not as intimate as in our problems. None of these models is applicable to the scheduling models we study. The few existing models that do address joint scheduling decisions of production and distribution are either special cases of our models or have a different structure from our problems. None of these models considers production costs. Many of them consider only time based performance measure in the objective function without taking into account any associated production or transportation costs. Most existing results that integrate production with transportation activities consider one of the following two special cases: (i) transportation costs are assumed to be zero, and hence the objective is optimize a job performance only; (ii) transportation times are assumed to be zero, i.e. job delivery can be done instantaneously. Potts (1980), Hall and Shmoys (1992), and Woeginger (1994, 1998) study a model in which orders are first processed in a single plant and then delivered to their customers. The objective is to minimize the maximum order lead time. Since transportation cost is not considered as a part of the objective in their model, each order is delivered as a separate shipment immediately after it is processed. Hence distribution scheduling is trivial, and production scheduling is the only decision to make. Lee and Chen (2001) and Li et al. (2005) study various problems of minimizing the maximum or total completion time of orders subject to the constraint that there are a limited number of transporters available for job delivery. Because of this restriction, a number of orders may have to be delivered together in a single shipment. Hall et al. (2001) investigate a similar model with the restriction that there are a fixed set of delivery dates at which the completed orders can be delivered. Herrmann and Lee (1993), Chen (1996), Yuan (1996), Cheng et al. (1996), Wang and Cheng (2000), and Hall and Potts (2003) consider a different set of models that treat both delivery lead time and transportation cost as part of the objective, but assume that the order delivery is done instantaneously without any transportation time. The lead time performance is measured by total weighted delivery earliness and tardiness of orders in the problems studied by Herrmann and Lee, Chen, Yuan, and Cheng et al. The problems considered by Wang and Cheng and Hall and Potts have different structures than ours.

The only paper that studies problems with delivery lead time and transportation cost as part of the objective function and with nonzero delivery times is by Chen and Vairaktarakis (2005). However, as in the problems studied in all of the above-cited papers, in their problems production costs are not considered and all the orders are processed in a single plant and hence any subset of jobs can be delivered in the same shipment as long as the total size does not exceed the shipment capacity. As we will note later, several special cases of our first model reduce to some of the problems considered by Chen and Vairaktarakis.

Scheduling problems with maximum tardiness related objectives have been studied extensively in the machine scheduling area (e.g. Pinedo 2002). However, most of the existing studies in this area consider only production operations. One of the earliest results is the EDD rule (Jackson, 1955) that minimizes the maximum tardiness by scheduling the orders in the non-decreasing order of their due dates on a single machine. When we consider order delivery along with order processing, we may want to batch together a set of orders for shipping in order to reduce the total distribution cost. So batching becomes important. Even if an order in a batch is processed early, it has to wait till all the other orders in the batch are processed before getting delivered. Webster and Baker (1995) and Potts and Kovalyov (2000) provide an extensive review of research in the area of scheduling with batching. However, many of the models described there differ from our model since batching in those models is done to take care of setup times between orders from different families instead of distribution. These problems deal only with the production part and do not consider production-delivery integration. For example, dynamic programming based and branch and bound based algorithms have been proposed to minimize the maximum tardiness for the single machine case with batch setup times (Ghosh and Gupta 1997, Hariri and Potts 1997).

One of the models studied in this thesis integrates production with distribution operations where each order may have different delivery capacity requirements. As we will see, the added packing decision makes the problems more challenging and requires different solution approaches than the models that assume the same delivery weight for each order. We consider various cases where an order may or may not be split for processing or delivery. Both cases of non-splittable and splittable order processing are widely considered in production scheduling literature (see, e.g. Pinedo 2002). While most distribution and routing models in the literature consider the case of non-splittable order delivery, there are a few models (see, e.g. Dror and Trudeau 1989, Belenguer et al. 2000) that consider the case of splittable order delivery. We are aware of only one production-distribution scheduling paper (Chang and Lee 2004) that assumes that each order has a generally different weight, thus incorporating order packing as a part of the scheduling decision. However, in the model considered by Chang and Lee, there is only a single delivery vehicle available to deliver all the orders. So the vehicle may not be available to deliver a batch of orders even if all the orders in it have completed processing because the vehicle has to return to the processing facility after delivery in order to pick up the next delivery batch. Also the objective in the Chang and Lee's model is to optimize a delivery time related performance without considering delivery costs. Chang and Lee study several problems by proposing heuristics for them and analyze the worst-case performance of the heuristics. In our model, there is no limit on the number of delivery vehicles and each batch is delivered by a separate vehicle immediately after the orders in it have completed processing.

The models that consider production time and capacity constraints can be divided into two broad classes based on how the demand is modeled. One class of models deals with dynamic demand patterns over a finite planning horizon and seeks to find a joint dynamic production and distribution schedule at a minimum total cost over the planning horizon. Recent publications in this area include Dogan and Goetschalckx (1999), Sabri and Beamon (2000), Jayaraman and Pirkul (2001), and Kaminsky and Simchi-Levi (2003). Because the demand varies with time, it is unlikely that closed-form optimal solutions exist for this class of models. Often, mathematical programming based solution approaches are used. The other class of models assumes constant production and demand rates and an infinite planning horizon, and seeks to find a joint cyclic production and delivery schedule at a minimum total cost per unit time. Closed-form optimal solutions may be available for this class of models under some policies on the relationship between the production and delivery cycles. Since the third model we study belongs to the second class of models mentioned here, we provide a detailed review of the related literature in this area in the following paragraphs.

All existing models in this area involve a single-stage supply chain consisting of one or more suppliers producing products and one or more customers ordering products directly from the supplier(s) without going through a warehouse. Most models are variations of the one-supplier-one-customer model where a single product is produced at a single supplier and delivered directly from the supplier to the customer and the production, transportation and inventory characteristics are the same as in our model. Most models are concerned with finding an optimal cyclic schedule from a given class of policies. The following two classes of policies are commonly considered:

a) Production cycle time and delivery cycle time are identical.

b) Production cycle time is an integer multiple of delivery cycle time.

Hahm and Yano (1992) consider the one-supplier-one-customer model mentioned above. They assume that the unit inventory holding cost at the supplier is the same as that at the customer. They show that production and delivery cycles in the optimal solution satisfy policy (b) and formulate the problem as a nonlinear mixed integer program which is solved by a heuristic approach. Benjamin (1989) studies the same problem except that the inventory cost at the supplier is calculated differently than Hahm and Yano (1992). The model studied by Hahm and Yano (1992) is extended by Hahm and Yano (1995a, 1995b, 1995c) to include multiple products, each with a constant production and demand rate. Jensen and Khouja (2004) give a polynomial time algorithm that can find the optimal solution for the same problem studied by Hahm and Yano (1995a).

Single-stage models with multiple suppliers or/and multiple customers are studied by Benjamin (1989), Blumenfeld et al (1985, 1991), and Hall (1996). Benjamin (1989) considers a model with multiple suppliers and multiple customers where only a single product is involved. The objective is to determine a cyclic production schedule at each supplier, and a cyclic delivery schedule for each transportation link between each supplier and each customer. It is formulated as a nonlinear program for which a heuristic solution procedure is designed. Blumenfeld et al (1985) consider various delivery options from suppliers to customers including direct shipping, shipping via a consolidation terminal, and a combination of terminal and direct shipping. Problems with one or multiple suppliers and one or multiple customers are considered under various assumptions. Blumenfeld et al (1991) study a model with one supplier and multiple customers where the supplier produces multiple products, one for each customer. Each product is allowed to be produced multiple times within a production cycle. In the case when all the products are homogeneous (i.e. have identical parameters), the production cycle is identical for all the products, and there is an identical number of production runs for each product within a production cycle, the authors derive the optimal production and delivery cycle times under policy (b). Hall (1996) considers various scenarios: one or more suppliers, one or more customers, one or more machines at each supplier, and one or more products that can be processed by each machine. He derives the cost formulas for many scenarios under policy (a).

Our model is more complex and more general in structure than the models considered in the above-reviewed literature because our model involves a two-stage supply chain whereas all of the existing models involve a single-stage supply chain. Although some of the production and delivery characteristics in our model are similar to some of the existing models, our model is in general more difficult to solve because of the added complexity of the warehouse in the supply chain. Furthermore, as discussed later, the study of this two-stage supply chain enables us to evaluate the value of warehouse in the supply chain and obtain related managerial insights.

The structure of our third model may also be viewed as a multistage assembly system if we view the warehouse as an assembly stage. In this context, the production (i.e. assembly) at the warehouse would be instantaneous because it does not really assemble the products; it merely puts all the products together for joint delivery to the customer. Therefore, our model may be viewed as a special lotsizing model for a multistage assembly system. In the following, we compare our model and solution approaches with existing ones in the area of lotsizing for multistage assembly systems with an infinite planning horizon and constant demand. First of all, to our knowledge, none of the existing lotsizing models for multistage assembly systems explicitly consider delivery from stage to stage (i.e. products are transferred from stage to stage at zero cost), and none of them consider production setup times and hence production capacity constraints due to setup times.

To see other differences, we consider existing models with a finite production rate at each facility separately from existing models with an infinite production rate (i.e. instantaneous production) at each facility. Comparing to the existing models with an infinite production rate at each facility (Crowston, et al. 1973, Blackburn and Millen 1982, Moily and Matthews 1987), our model has different and more complex inventory functions at the suppliers because the production rates at the suppliers in our model are finite, which leads to the requirement of inventory accumulation prior to each delivery to the warehouse. Given this and the fact that there are capacity constraints in our model but not in those existing models, the solution approaches used in those papers cannot be applied to our model.

All the existing models with a finite production rate at each facility (Schwarz and Schrage 1975, Moily 1986, Atkins et al. 1992) assume that the production rates are non-increasing across the system (i.e. from components to final products), whereas this assumption does not hold in our model if our model is viewed as a multistage assembly system. Crowston, et al. (1973) show the property that under this assumption and without the capacity constraint due to setup times, the lot size at each facility is an integer multiplier of that at each immediately succeeding facility. This property is similar to one of the results we prove in our model. However, our result is proved without this assumption and with the capacity constraint. Schwarz and Schrage (1975) use this property to formulate the problem as an integer program. They propose a branch-and-bound algorithm for getting optimal solutions and a heuristic procedure that optimizes the system as a collection of two-stage systems by ignoring multistage interaction effects. In addition to the non-increasing-production-rates assumption, the models studied by Moily (1986) and Atkins et al. (1992) assume that the product is transferred from one stage to the next immediately and continuously upon its completion, whereas in our model all the units in a delivery shipment are transferred together. Because of this difference, how inventory accumulates and hence the inventory function at various facilities in our model are different from the models considered in these existing papers. Atkins et al. (1992) derive some theoretical results for which the non-increasing-production-rates assumption is a key. The solution approach used by Moily (1986) is different from the approach we use. His approach is based on a one-time rounding of any non-integer multipliers obtained without taking into account its effects on the other participants of the system.

As quick response is becoming more critical in many supply chains, the linkage between production and distribution is becoming ever more intimate. Consequently, joint consideration of order processing and delivery scheduling is becoming crucial in achieving quick response at minimum cost. Because of this growing importance, an increasing amount of research has been devoted to integrated production-distribution scheduling models in the last several years. However, this area is relatively new and more research is needed. The models we study in this paper contribute to this area by analyzing various production-distribution scheduling models.

1.6 Summary

The objective of this work is to study integrated production and distribution scheduling decisions in various supply chains. While a lot of literature exists on exclusive production scheduling or distribution scheduling, our study shows that optimizing these performance measures independently may lead to a suboptimal system solution. With increasing competition, supply chain optimization as opposed to individual operation optimization becomes crucial. This study aims to provide numerous insights and approaches to implement supply chain scheduling decisions integrating production and distribution operations.

In Chapters 2 through 5, we cover the four different supply chain models. In Chapter 2, we consider a setup with multiple manufacturing plants owned by the same firm where the firm has to decide on the order allocation and production and distribution scheduling. Chapter 3 deals with the make-to-order production-distribution system with one supplier and one or more customers where we look at due dates and distribution costs. In Chapter 4, we study an integrated production and distribution scheduling model in a two-stage supply chain to find a cyclic production schedule at each supplier, a cyclic delivery schedule from each supplier to the warehouse, and a cyclic delivery schedule from the warehouse to the customer so that the customer demand for each product is satisfied fully at minimum total production, inventory and distribution cost. Chapter 5 combines order processing with packing decisions for delivery in a make-to-order supply chain with one supplier and one customer. Chapter 6 gives the conclusions and scope for further work.

Chapter 2

Order Assignment and Scheduling in a Supply Chain

2.1 Introduction

Globalization has become a competitive strategy for many manufacturing firms due to the cheaper labor and raw material costs overseas. About a fifth of the output of American companies is produced abroad and around 53% of American firms are multinational (Dornier et al. 1998). The supply chain of a typical American multinational manufacturer may consist of a number of plants located at several foreign countries and a central distribution center (DC) in the United States where products are received from overseas plants and distributed to many domestic retail stores. In such a supply chain, production costs and productivity may vary significantly from plant to plant due to variations in labor costs and skills in the different countries. Also, in such a supply chain, transportation costs are generally higher, and distribution lead times longer than in a domestic supply chain.

Now consider the global supply chain of a manufacturer who produces time-sensitive products, such as toys, fashion apparel, or high-tech products that typically have a large variety, a short life cycle, and are sold in a very short selling season (Hammond and Raman 1996, Johnson 2001). Because of high demand uncertainty of the products, retailers typically do not place orders until reliable market information is available shortly before a selling season. On the other hand, since there are significant markdowns for unsold products at the end of the selling season, the manufacturer runs a high risk if it starts production early before it receives orders from the retailers. As a result, the manufacturer would not start production until orders from the retailers have been placed shortly before the selling season. Due to the fact that there is only a limited amount of production time available, in order to deliver the orders to the retailers as soon as possible at a low cost, the manufacturer has to schedule the production and distribution operations in a coordinated and efficient manner. In this chapter we consider a simplified version of the order assignment and scheduling problem faced by the manufacturer in the above-described supply chain. In this problem, the manufacturer receives a set of distinct orders from the retailers before a selling season, and needs to determine (i) which orders to be assigned to which plants, (ii) how to schedule the production of the assigned orders at each plant, and (iii) how to schedule the distribution of the completed orders from each plant to the DC, so as to optimize a certain performance measure of the supply chain. Due to the variations in productivity and labor costs at different plants, the processing time and cost of an order are dependent on the plant to which it is assigned. Completed orders are delivered in shipments from the plants to the DC. Each shipment can carry up to a certain number of orders and is associated with a certain distribution time and a certain distribution cost. Since the products are time-sensitive, an important factor related to the performance of the supply chain is the delivery lead time, i.e. the time between the placement of an order by a retailer and its delivery to the retailer. We assume that the DC is located close to the retailers, such that the delivery time and cost from the DC to the retailers are negligible, compared to the delivery time and cost from the plants to the DC. Therefore, the lead time of an order in our problem is the time between the placement of the order and its delivery to the DC. Another important factor related to the performance of the supply chain is cost. The total cost in this supply chain consists of production costs for processing the orders at the plants and distribution costs for the delivery of completed orders from the plants to the DC. Since finished products are rarely held at the plants or DC for a long time in such a time-sensitive supply chain, inventory cost of finished products is negligible and not considered. We consider four different performance measures of the supply chain, each of which takes into account both the delivery lead time and the total cost. A problem corresponding to each performance measure is studied separately. The problems we study integrate order assignment, production scheduling (for order processing at the plants), and distribution scheduling (for the delivery of completed orders from the plants to the DC).

In the broader literature of supply chain management, a tremendous amount of research has been done on various strategic and tactical problems in the past decade. However, very few results have addressed scheduling issues in a supply chain. On the other hand, as quick response is becoming more and more critical in many manufacturing and service supply chains, the linkage between production and distribution is becoming ever more intimate. Consequently, optimal scheduling of orders across different stages of a supply chain is becoming crucial in achieving quick response at minimum cost. This chapter has two objectives. Our first objective is to analyze the computational complexity of various cases of the problems we consider by either proving that a problem is intractable (i.e., NP-hard) or providing an efficient exact algorithm for the problem. Our second objective is to design fast heuristics for NP-hard problems that are capable of generating near optimal solutions. We evaluate the performance of the heuristics by analyzing their worst-case and asymptotic performances and conducting computational experiments. This chapter is organized as follows. In Section 2.2, we specify the notation, define the problems, and give some optimality properties of the problems. We then study the problems in Sections 2.3 through 2.6, respectively. Finally, in Section 2.7 we conclude the chapter.

2.2 Problems and Preliminary Results

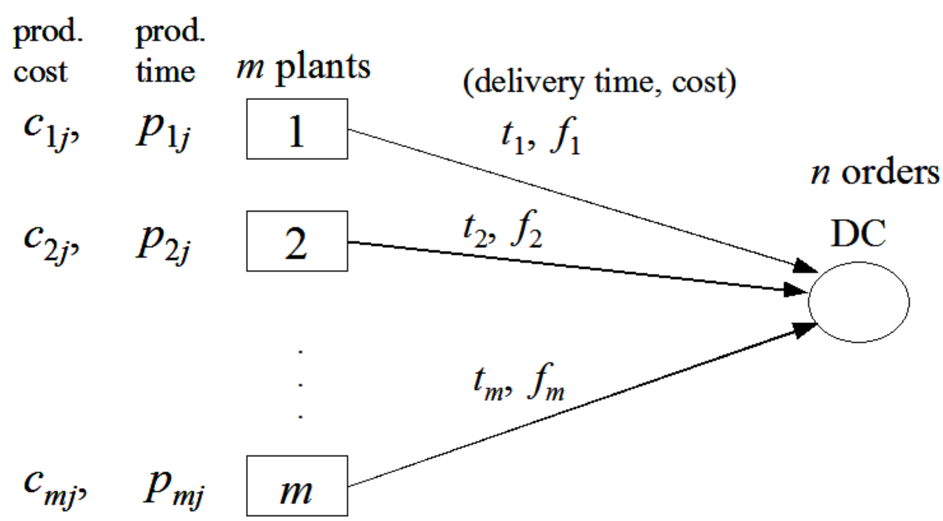

Figure 2.1: The supply chain

In this section we define our problems mathematically and introduce some optimality properties satisfied by all the problems which we will use in later sections. A schematic diagram of the supply chain is given in Fig 2.1. We are given n customer orders, $N = \{1, 2, \ldots, n\}$, at time 0, each of which is to be processed at one of m plants in a supply chain, $M = \{1, 2, \ldots, m\}$. Each plant has a single dedicated production line and is capable of producing all the orders. It takes p_{ij} units of processing time and c_{ij} units of production cost for plant i to process order j, for $i \in M$ and $j \in N$. Each order only needs to be processed by one of the plants once without interruption. Completed orders are delivered to a distribution center (DC) in the supply chain. The delivery time and delivery cost of a shipment from plant $i \in M$ to the DC are t_i and f_i, respectively. Each delivery shipment has a capacity limit; it can carry up to b orders. We assume that each order takes up the same amount of capacity of a shipment and that partial delivery of an order is not possible. The problem is to assign each order to a plant, schedule the processing of the orders assigned to each plant, and schedule the delivery of the completed orders from each plant to the DC, so as to optimize a given objective function that takes into account delivery lead time, total production cost, and total distribution cost. To schedule the processing of assigned orders at each plant, we need to determine which sequence of the orders to use and when to start processing each order. Similarly, to schedule the delivery of the completed orders, we need to determine how many shipments to use at each plant, which orders to be delivered in each shipment, and when each shipment should depart from the plant. For a given schedule, we define:

TC: the total cost of production and distribution

C_j: the completion time of order $j \in N$ which is the time when order j completes processing at the plant to which it is assigned.

D_j : the delivery time of order $j \in N$ which is the time when order $j \in N$ is delivered to the DC.

Since all the orders are given at time 0, D_j also represents the lead time of order j. We consider the following two functions for measuring the delivery lead time performance of the supply chain:

(i) total lead time of the orders, $D_{\text{total}} = \sum_{j \in N} D_j$.

(ii) maximum lead time of the orders, $D_{\max} = \max\{D_j | j \in N\}$.

These two functions are analogous to two widely used functions for measuring customer service in the production scheduling literature (e.g. Pinedo 2002), total completion time $C_{\text{total}} = \sum_{j \in N} C_j$, and maximum completion time $C_{\max} = \max\{C_j | j \in N\}$. In the traditional scheduling literature, it is implicitly assumed that once an order completes processing it is delivered to its customer immediately without any transportation time or cost, and hence C_j is treated as the lead time of order j. However, in our problems, since transportation cost is considered, an order may be delivered together with some other orders and hence it may not be delivered immediately after it is processed. Moreover, there are transportation times in our problems. Hence $D_j > C_j$ and D_j, instead of C_j, is the lead time of order j.

We consider the following four problems, each with a different objective:

P1: Minimizing a weighted sum of the total lead time and total cost, i.e. $\alpha D_{\text{total}} + (1-\alpha)TC$, where $\alpha \in [0,1]$ is a given constant, representing the decision-maker's relative preference on D_{total} and TC.

P2: Minimizing the total cost TC subject to the constraint that the total lead time is no more than a given threshold, i.e. $D_{\text{total}} \leq D$, where D is a given constant.

P3: Minimizing a weighted sum of the maximum lead time and total cost, i.e. $\alpha D_{\max} + (1-\alpha)TC$, where $\alpha \in [0,1]$ is a given constant, as in problem P1.

P4: Minimizing the total cost TC subject to the constraint that the maximum lead time is no more than a given threshold, i.e. $D_{\max} \leq D$, where D is a given constant.

We note that several special cases of the problems P1 and P3 are related to some existing scheduling problems in the literature. The special case of P1 and P3 with a single plant (i.e. $m = 1$) are equivalent to two of the problems studied by Chen and Vairaktarakis (2004). They give polynomial-time algorithms for finding optimal solutions to those problems. In the single-plant case, there are no order assignment decisions to be made and production costs can be ignored, and hence the problems are much easier. However, as we will see later, the general cases of P1 and P3 with multiple plants are NP-hard. In this chapter, we focus on the multi-plant problems only. If we view each plant as an unrelated parallel machine and assume zero production costs and zero delivery times and costs, then P1 and P3 reduce to the classical unrelated parallel machine scheduling problems with total completion time and maximum completion time of orders as the objective function, respectively. It is known that the unrelated parallel machine total completion time problem can be formulated as an assignment problem and solved in polynomial time (Horn 1973), and the unrelated parallel machine maximum completion time problem is NP-hard (Garey and Johnson 1979).

We say that a set of orders assigned to some plant i are in SPT order (shortest-processing-time-first order) if they are sequenced in the non-decreasing order of their processing times p_{ij} and orders with equal processing times are sequenced in the same order as their indices. In the following, we present some preliminary results about the structure of an optimal schedule.

Lemma 1 There exists an optimal schedule for all the problems P1, P2, P3, and P4 in which all of the following hold: (1) The orders assigned to each plant are scheduled in the SPT order, (2) There is no inserted idle time between orders processed at each plant, (3) The departure time of each shipment is the time when all the orders in it complete processing, (4) All the orders that are delivered in the same shipment are processed consecutively at a plant.

Proof (1) If any order violates this rule, we can rearrange the orders in the SPT order without increasing the objective function value. If the violating orders are in the same shipment, there will not be any change in the value of the objective function as each order has to wait for the other orders in the shipment before it gets delivered, and hence the sequence of orders within a batch does not matter. If the orders that violate this rule are in different shipments, then after re-sequencing the orders in the SPT order, we can adjust each shipment such that it consists of the jobs at the same positions in the new sequence as in the original sequence. This will lead to a decrease in the departure times of some shipments and hence a decrease in the objective value. (2), (3), and (4) can be proved easily. We omit the proofs for them. ■

Lemma 2 There exists an optimal schedule for all the problems P1, P2, P3, and P4 in which the number of orders delivered in an earlier shipment from a plant is greater than or equal to the number of orders delivered in a later shipment from the same plant.

Proof Consider two consecutive shipments S_1 and S_2 from some plant $i \in M$, where S_1 is delivered earlier than S_2. Suppose that there are n_1 and n_2 orders in S_1 and S_2, respectively, such that $n_1 < n_2$. Let u_1 and u_2 denote the completion time of the last order in S_1 and S_2 respectively. The contribution of the orders in S_1 and S_2 to the total lead time D_{total} is thus given by

$$F(S_1, S_2) = n_1(u_1 + t_i) + n_2(u_2 + t_i)$$

Now we move the first order in S_2 to S_1. Let the processing time of this order be p. Then the contribution of the orders in S_1 and S_2 to D_{total} becomes

$$G(S_1, S_2) = (n_1 + 1)(u_1 + p + t_i) + (n_2 - 1)(u_2 + t_i)$$

Since no other shipments are involved, the total contribution to D_{total} by the orders in the shipments other than S_1 and S_2 remains the same. Therefore, the value of D_{total} is decreased by

$$F(S_1, S_2) - G(S_1, S_2) = u_2 - u_1 - (n_1 + 1)p$$

By Lemma 1(1), we can assume that all the orders in S_1 and S_2 are in SPT order. Thus the processing time of each order in S_1 is no more than p, whereas that of each order in S_2 is at least p. By the assumption that $n_1 < n_2$, we have $u_2 - u_1 \geq n_2 p \geq (n_1 + 1)p$. This means that $F(S_1, S_2) - G(S_1, S_2) \geq 0$, i.e. the value D_{total} is not increased after moving the first order of S_2 to S_1. Clearly, the values $D_{\max}$ and TC both remain the same. Therefore, the objective value of each of the problems P1, P2, P3, P4 is not increased after moving the first order of S_2 to S_1. We can repeat this until the number of orders in S_1 is equal to that in S_2. ■

2.3 Problem P1: Minimizing $\alpha D_{\text{total}} + (1 - \alpha)TC$

We first discuss two extreme cases of the problem with $\alpha = 1$ or $\alpha = 0$. We then show that the general problem P1 with $0 < \alpha < 1$ is NP-hard, propose two heuristics for the problem, and evaluate both theoretical and computational performance of the heuristics. The following two special cases of problem P1 arise in many practical situations and hence are considered separately: (i) the order processing times are agreeable, i.e. there exists an ordering of the orders, denoted as $([1], \ldots, [n])$ which is a permutation of $(1, \ldots, n)$, such that $p_{i[1]} \leq \ldots \leq p_{i[n]}$, for all $i \in M$; (ii) the production costs of orders at each plant are proportional to the processing times of the orders, i.e. $c_{ij} = \gamma_i p_{ij}$ for $i \in M$ and $j \in N$, where γ_i represents the production cost per unit processing time at plant i. We give dynamic programming algorithms with a time complexity polynomial in n and exponential in m for these problems.

2.3.1 Problem P1 with $\alpha = 0$ or $\alpha = 1$

In problem P1 with $\alpha = 1$, since no cost is considered, each order is delivered in a separate shipment immediately after it completes processing. This case of the problem can be formulated as an assignment problem as follows. For $k, j \in N$, and $i \in M$, define a parameter $a_{(k,i)j} = kp_{ij} + t_i$ which is the contribution to D_{total} by order j if it is scheduled to the kth last position at plant i. Define a binary variable $x_{(k,i)j}$ to be 1 if order j is scheduled as the kth last order at plant i, and 0 otherwise. The following assignment problem formulates problem P1 with $\alpha = 1$.

$$\min \sum_{k \in N} \sum_{i \in M} \sum_{j \in N} a_{(k,i)j} x_{(k,i)j}$$

Subject to:

$$\sum_{k \in N} \sum_{i \in M} x_{(k,i)j} \quad = \quad 1 \quad j \in N$$

$$\sum_{j \in N} x_{(k,i)j} \quad \leq \quad 1 \quad k \in N \quad i \in M$$

$$x_{(k,i)j} \in \{0, 1\} \qquad k \in N, i \in M, j \in N$$

It is well-known that solving the LP relaxation of this formulation yields an integer solution. Thus problem P1 with $\alpha = 1$ is solvable in polynomial time.

Problem P1 with $\alpha = 0$ is to minimize the total production and distribution cost. This problem can be solved by the following procedure which has a time complexity polynomial in n and exponential in m. Since D_{total} is not considered, each delivery shipment at each plant should deliver as many orders as possible in order to minimize the total transportation cost. Suppose that there are n_i orders assigned to plant $i \in M$ in an optimal solution. Then $\lceil \frac{n_i}{b} \rceil$ shipments are used at plant i and hence the total distribution cost is fixed as $\sum_{i \in M} \lceil \frac{n_i}{b} \rceil f_i$. The problem is then reduced to minimizing the total production cost, which can be formulated as the following transportation problem. Define x_{ij} to be 1 if order j is assigned to plant i, and 0 otherwise.

$$\min \sum_{i \in M} \sum_{j \in N} c_{ij} x_{ij}$$

Subject to:

$$\begin{aligned} \sum_{i \in M} x_{ij} &= 1 \quad j \in N \\ \sum_{j \in N} x_{ij} &= n_i \quad i \in M \quad i \in M \\ x_{ij} \in \{0,1\} & \qquad i \in M, j \in N \end{aligned}$$

Solving the LP relaxation of this formulation gives an integer solution. Therefore, for a given combination of $(n_1, \ldots, n_m)$, problem P1 with $\alpha = 0$ can be solved in polynomial time. We can enumerate all possible combinations of $(n_1, \ldots, n_m)$ with $n_1 + \ldots + n_m = n$, and for each combination solve such a transportation problem. The solution with the lowest total production and distribution cost is then optimal to problem P1 with $\alpha = 0$. Since there are no more than n^m possible combinations of $(n_1, \ldots, n_m)$ with $n_1 + \ldots + n_m = n$, the above procedure is polynomial for problem P1 with $\alpha = 0$ and a fixed m. However, when the number of plants m is arbitrary, problem P1 with $\alpha = 0$ becomes NP-hard, which is proved in the following theorem.

Theorem 1 Problem P1 with $\alpha = 0$ and an arbitrary number of plants is strongly NP-hard.

Proof We prove this by a reduction from the Minimum Cover (MC) problem, which is known to be strongly NP-complete (Garey and Johnson 1979).

MC: Given a set S with h elements $S = \{1, \ldots, h\}$, a collection Q of u subsets of S, $Q = \{S_1, \ldots, S_u\}$, where S_i is a subset of S, for $i = 1, \ldots, u$, and a positive integer $v \leq u$, does there exist a subset Q' of Q with $|Q'| \leq v$, such that every element of S belongs to at least one member of Q'?

Given this instance of MC, we consider an instance of the recognition version of P1 defined by:

Number of orders, $n = h$, and set of orders, $N = S$

Number of plants, $m = u$, and set of plants, $M = \{1, \ldots, u\}$.

Order processing times, $p_{ij} = 0$, for $i \in M$ and $j \in N$.

Order production costs, $c_{ij} = 0$ if $j \in S_i$, and $2u$ otherwise, for $i \in M$ and $j \in N$.

Delivery times, $t_i = 0$, and delivery costs, $f_i = 1$ for $i \in M$.

Shipment capacity, $b = h$.

Threshold of objective value, $Z = v$.

We show that there is a schedule to this instance of P1 with the objective value no more than Z if and only if there is a solution to MC.

(If part) Without loss of generality, we assume that $Q' = \{S_1, \ldots, S_w\}$ with $w \leq v$ is a solution to MC. We construct a solution to P1 as follows. For each $j \in N$, define $P_j = \{i \in \{1, \ldots, w\} | j \in S_i\}$, and assign order j to any plant $i \in P_j$. Since every element of S is covered by S_i for some $i \in \{1, \ldots, w\}$, every order $j \in N$ gets assigned to a plant in M. Use one shipment to deliver all the orders assigned to each plant. This gives a solution to P1. Since order j is assigned to a plant i with $j \in S_i$, the production cost of order j is 0. Thus the total production and distribution cost of this solution is no more than $w \leq Z$.

(Only If part) Given a solution to P1 with the total production and distribution cost no more than Z, we can conclude that all the orders are assigned to plants where their production costs are zero. This is because if an order was assigned to a plant with a positive production cost, then the total cost would be more than $2u > Z$. Let k be the number of plants where orders are assigned. Clearly, $k \leq v$. Without loss of generality, suppose that all the orders are assigned to plants $1, \ldots, k$. If order j is assigned to plant $i \in \{1, \ldots, k\}$, then $j \in S_i$ because otherwise c_{ij} would be nonzero. This means that $\{S_1, \ldots, S_k\}$ is a solution to MC.

■

2.3.2 Problem P1 with Agreeable Processing Times

In many practical situations, given a set of orders, there is a clear ordering with respect to their processing times, regardless of which plant they are processed. For example, if order 1 requires more time than order 2 if processed at one plant, it is likely to be the same case at every other plant. In this case of the problem, we say that the order processing times are agreeable, i.e. there exists an ordering of the orders, denoted as $([1], \ldots, [n])$ which is a permutation of $(1, \ldots, n)$, such that $p_{i[1]} \leq p_{i[2]} \leq \ldots \leq p_{i[n]}$, for all $i \in M$.

We give a dynamic programming algorithm to solve P1 with agreeable processing times. We say that a set of orders assigned to plant i are in LPT order (longest-processing-time-first order) if they are sequenced in the non-increasing order of their processing times p_{ij} and orders with equal processing times are sequenced in the reverse order of their indices. Since the processing times are agreeable, both the SPT and LPT order of a given set of orders remains the same regardless of which plant they are assigned to. This property enables us to use a common sequence of the orders in the dynamic program. Our DP algorithm considers the orders in LPT order and assigns them to a plant backward from the last position to the first. The resulting forward sequence of the orders assigned to a plant by this algorithm is thus in SPT order, satisfying Lemma 1 (1).

Algorithm DP-P1A

Re-index the orders such that the order indices $(1, \ldots, n)$ are in the LPT order.

Define value function $F(j; j_1, \ldots, j_m; b_1, \ldots, b_m; h_1, \ldots, h_m)$ to be the minimum total contribution to the objective function by the first j orders from the LPT order, given that there are j_i orders scheduled backward at plant i, there are h_i orders already scheduled in the current earliest shipment at plant i, and there will be b_i orders in the current earliest shipment at plant i in the final schedule.

Initial values

$F(0; 0, \ldots, 0; 0, \ldots, 0; 0, \ldots, 0) = 0$

$F(j; j_1, \ldots, j_m; b_1, \ldots, b_m; h_1, \ldots, h_m) = \infty$, for each state $(j; j_1, \ldots, j_m; b1, \ldots, b_m; h_1, \ldots, h_m)$ that violates at least one of the following conditions: $j_1 + \ldots + j_m = j$; $1 \leq b_i \leq b$ and $1 \leq h_i \leq \min(b_i, j_i)$, for $i \in M$.

Recursive relations

For each state $(j; j_1, \ldots, j_m; b_1, \ldots, b_m; h_1, \ldots, h_m)$ satisfying all of the following conditions: $j_1 + \ldots + j_m = j$; $1 \leq b_i \leq b$ and $1 \leq h_i \leq \min(b_i, j_i)$, for $i \in M$,

$F(j; j_1, \ldots, j_m; b_1, \ldots, b_m; h_1, \ldots, h_m) = \min\{i \in M\}$

$$\begin{cases} F(j-1; j_1, \ldots, j_i - 1, \ldots, j_m; b_1, \ldots, b_m; h_1, \ldots, h_i - 1, \ldots, h_m) + \alpha[(j_i - h_i + b_i)p_{ij} + t_i] \\ \qquad\qquad + (1-\alpha)c_{ij}, \text{ if } h_i \geq 2 \\ \min_{\{1 \leq b_i \leq b\}}\{F(j-1; j_1, \ldots, j_i - 1, \ldots, j_m; b_1, \ldots, b_i', \ldots, b_m; h_1, \ldots, b_i', \ldots, h_m) \\ \qquad\qquad + \alpha[(j_i - 1 + b_i)p_{ij} + t_i] + (1-\alpha)(c_{ij} + f_i)\}, \text{ if } h_i = 1 \end{cases}$$

Optimal solution

An optimal solution is provided by minimizing $F(n; j_1, \ldots, j_m; b_1, \ldots, b_m; b_1, \ldots, b_m)$ over all the states $(n; j_1, \ldots, j_m; b_1, \ldots, b_m; b_1, \ldots, b_m)$ with $j_1 + \ldots + j_m = n$.

We note that in algorithm DP-P1A, the value function $F(j; j_1, \ldots, j_m; b_1, \ldots, b_m; h_1, \ldots, h_m)$ is obtained by assigning order j to the plant in M that results in the minimum total contribution to the objective function. Since the algorithm schedules the orders assigned to each plant backward from the last position to the first, if order j is assigned to plant i, then the following two cases specify the exact contribution of order j to the objective function:

(1) If the order is added to the current earliest shipment with final size b_i (in this case, $h_i \geq 2$), then its contribution is $\alpha[(b_i + j_i - h_i)p_{ij} + t_i] + (1-\alpha)c_{ij}$. The term $(b_i + j_i - h_i)p_{ij}$ is because order j contributes p_{ij} units of time to the delivery time of each order in the current earliest shipment and each order after that shipment (i.e. a total of $b_i + j_i - h_i$ orders).

(2) If the order is added to a new shipment with the final size b_i before the current earliest shipment (in this case, $h_i = 1$) , then its contribution is $\alpha[(b_i + j_i - h_i)p_{ij} + t_i] + (1-\alpha)(c_{ij} + f_i)$.

Theorem 2 Algorithm DP-P1A solves problem P1 with agreeable processing times to optimality in $O(n^{m+1}mb^{2m})$ time.

Proof The recursive relations of the dynamic program cover all possible state transitions and hence the optimality of the algorithm is guaranteed. In the value function, j can range from 1 to n, the maximum number of combinations of $(j_1, \ldots, j_m)$ with $j_1 + \ldots + j_m = j$ is bound by n^m, and each b_i is bound by b. Hence there are no more than $n^{m+1}b^m$ possible combinations of $(j, j_1, \ldots, j_m; b_1, \ldots, b_m)$ in the dynamic program. Let $u(q)$ denote the number of combinations of $(h_1, \ldots, h_m)$ where the number of h_i's with a value 1 is exactly $q (0 \leq q \leq m)$. Thus, there are no more than $u(q)n^{m+1}b^m$ states $(j, j_1, \ldots, j_m; b_1, \ldots, b_m; h_1, \ldots, h_m)$ where the number of h_i's with a value 1 is exactly q. Since each h_i varies from 1 to b, we can see that $u(q) = \binom{m}{q}(b-1)^{m-q}$. From the recursive relations, we can see that it takes $O(bq + m - q)$ time to calculate the value function for a state where the number of h_i's with a value 1 is exactly q. Therefore, the overall complexity of the algorithm is bounded by $O(y)$, where y is given as follows:

$$
\begin{aligned}
y &= n^{m+1}b^m \sum_{q=0}^{m} u(q)(bq + m - q) \\
&= n^{m+1}b^m \left[\sum_{q=0}^{m} \binom{m}{q} (b-1)^{m-q}(b-1)q + \sum_{q=0}^{m} \binom{m}{q} (b-1)^{m-q} m \right] \\
&= n^{m+1}b^m \left[\sum_{q=0}^{m} \binom{m}{q} (b-1)^{m-q}(b-1)q + mb^m \right] \quad (\text{since } \sum_{q=0}^{m} \binom{m}{q} (b-1)^{m-q} = b^m \ ) \\
&= n^{m+1}b^m(b-1) \left[\sum_{q=0}^{m} \binom{m}{q} (b-1)^{m-q} \right] + n^{m+1}mb^{2m} \\
&= n^{m+1}b^m(b-1) \left[\sum_{q=1}^{m} m \binom{m-1}{q-1} (b-1)^{m-q} \right] + n^{m+1}mb^{2m} \\
&= n^{m+1}b^m(b-1) \left[m \sum_{q=0}^{m-1} \binom{m-1}{q} (b-1)^{(m-1)-q} \right] + n^{m+1}mb^{2m} \\
&= n^{m+1}b^m(b-1)mb^{m-1} + n^{m+1}mb^{2m} \quad (\text{since } \sum_{q=0}^{m-1} \binom{m-1}{q} (b-1)^{(m-1)-q} = b^{m-1} \ ) \\
&\leq 2n^{m+1}mb^{2m}
\end{aligned}
$$

Therefore, the overall complexity is bounded by $O(n^{m+1}mb^{2m})$. ■

Theorem 2 means that if the number of plants m is fixed, problem P1 with agreeable processing times is solvable in polynomial time. However, it is unknown whether this case of the problem is NP-hard when the number of plants is arbitrary.

2.3.3 Problem P1 with Production Costs Proportional to Processing Times

In most production environments, the production cost of an operation is typically proportional to the time duration of the operation because both labor cost and resource consumption involved are usually proportional to the production time required by the operation.

Therefore, it is reasonable to assume that the production costs of orders at each plant are proportional to the processing times of the orders, i.e. $c_{ij} = \gamma_i p_{ij}$ for $i \in M$ and $j \in N$, where γ_i represents the production cost per unit processing time at plant i.

We show that problem P1 under this assumption can be solved to optimality by a dynamic programming algorithm which has a time complexity polynomial in n and exponential in m, meaning that the algorithm is polynomial if the number of plants m is fixed. Before presenting the algorithm, we first introduce some definitions to be used in the algorithm. Denote the SPT list of the orders with respect to their processing times at plant i by $SPT_i = ([i1], [i2], \ldots, [in])$, where $[ih]$ denotes the hth order in SPT_i. The m lists $SPT_1, \ldots, SPT_m$ are in general different because the processing times may not be agreeable. Define SPT_{iu} to be the set of the first u orders in SPT_i, i.e. $SPT_{iu} = \{[i1], \ldots, [iu]\}$.

For some given $k_1, \ldots, k_m$ with each $k_i \leq n$, if we know that all the orders in the joint set $Y_{i=1}^{m} SPT_{ik_i}$ have been scheduled, then we know exactly which orders are left in each list SPT_i, for $i \in M$. Note that some of the orders in $SPT_i \backslash SPT_{ik_i}$ may have been covered by the set $Y_{i=1}^{m} SPT_{ik_i}$. Thus the remaining orders in each list SPT_i is in general a subset of $SPT_i \backslash SPT_{ik_i}$. Given $k_1, \ldots, k_m$, and the fact that all the orders in $Y_{i=1}^{m} SPT_{ik_i}$ have been scheduled, we can know the remaining orders in each set SPT_i in polynomial time (polynomial in n and m). Similarly, for some given $q \in M$, if we know that all the orders in the joint set $Y_{i=1, i \neq q}^{m} SPT_{ik_i}$ have been scheduled, then we can know exactly which orders are remaining in SPT_q in polynomial time.

Our DP algorithm is based on the following result.

Lemma 3 There exists an optimal schedule $\pi = (\pi_1, \ldots, \pi_m)$, where π_i is the schedule at plant i, such that if we divide π arbitrarily into two parts, left part denoted as $\pi_L = (\pi_{1L}, \ldots, \pi_{mL})$ and right part denoted as $\pi_R = (\pi_{1R}, \ldots, \pi_{mR})$, where $(\pi_{iL}, \pi_{iR}) = \pi_i$, for every $i \in M$, then there exists some $i \in M$ such that $u_i = y_i$, where u_i is the first order in π_{iR}, and y_i is the first order in the SPT sequence of the orders in the set $\cup_{q\in M}\pi_{qR}$ with respect to the processing times at plant i.

Proof We prove this by showing that any schedule that violates this lemma can be transformed into a schedule that satisfies this lemma with an equal or lower total cost. Given a schedule π, suppose that there is a partition of $\pi = (\pi_L, \pi_R)$ such that under this partition $u_i \neq y_i$ for every $i \in M$. Define order set $Y = \{y_1, \ldots, y_m\}$. Let H denote the subset of plants where the orders from Y are scheduled. There are two cases:

Case 1: If $|H| = m$, then each plant processes exactly one order from the set Y. Let $y_{[i]}$ denote the order from Y which is processed at plant i, for $i \in M$. Create a new schedule by modifying π_R as follows: For $i \in M$, if $u_i = y_{[i]}$, then remove $y_{[i]}$ from π_{iR}, and move y_i from where it is scheduled in π_R to the position of $y_{[i]}$; otherwise, move y_i from where it is scheduled in π_R to the position right before u_i in π_{iR} so that y_i becomes the first order in the new π_{iR}. It can be easily verified that every order in this new schedule has a processing time no greater than that of the order scheduled in the same position in the original schedule. Since the production costs of the orders at each plant are proportional to their processing times, it can be seen that this new schedule has a total cost no more than that of the original schedule.

Case 2: If $|H| \leq m-1$, then define $Y_1 = \{y_i | i \in H\}$, and $H_1 = \{i \in H|$ plant i processes some order $y_j \in Y_1\}$. Clearly, $Y_1 \subseteq Y$ and $H_1 \subseteq H$. There are two cases again. If $|H| = |H_1|$, then each plant in H processes exactly one order from Y_1. Following the same approach as in Case 1, we can construct a new schedule where $y_i \in Y_1$ becomes the first order in π_{iR}, for every $i \in H_1$, and this new schedule has a total cost no more than that of π. If $|H| > |H_1|$, then follow the same argument as in Case 2 to further find a subset H_2 of

H_1 and a subset Y_2 of Y_1 such that $Y_2 = \{y_i | i \in H_1\}$, and $H_2 = \{i \in H_1|$ plant i processes some order $y_j \in Y_2\}$. Again there are two cases to consider: $|H_1| = |H_2|$ or $|H_1| > |H_2|$. Continue this process, and eventually we will have $|H_{q-1}| = |H_q|$ for some $q (q < m)$ and apply the same argument as in Case 1. ■

For $i \in M$, let SPT_{iR} denote the SPT sequence of the orders in the set $\cup_{q \in M} \pi_{qR}$ with respect to the processing times at plant i. Lemma 3 implies that we can build an optimal schedule step by step as follows. Suppose that we have built a partial schedule π_L. Next we can append the first order in the list SPT_{iR} to the end of π_{iL} for some plant $i \in M$. Since we do not know the exact plant $i \in M$ where we should add an order to, we can try every $i \in M$ (i.e. for every $i \in M$, we try to append the first order of SPT_{iR} to the end of π_{iL}), and select the resulting schedule with the lowest total cost. This observation enables us to develop a dynamic programming algorithm for the problem. The DP algorithm schedules the orders in SPT order at each plant forward from the first position to the last. Suppose that a partial schedule π_L has been built, the algorithm next tries to append the first order of SPT_{iR} to the end of π_{iL}, for every $i \in M$, and selects the resulting schedule with the lowest total cost. We describe the details of the algorithm below.

Algorithm DP-P1P

Define value function $F(n_1, \ldots, n_m; k_1, \ldots, k_m; b_1, \ldots, b_m; h_1, \ldots, h_m)$ to be the minimum total contribution of the orders in a partial schedule where:

(i) there are exactly n_i orders at plant i in the final schedule,

(ii) j_i orders have been scheduled currently at plant i,

(iii) the current last order scheduled at plant i is order $[ik_i]$,

(iv) the size of the current last shipment at plant i in the final schedule is b_i, and

(v) there are h_i orders already scheduled in the current last shipment at plant i.

Initial values

$F(n_1, \ldots, n_m; 0, \ldots, 0; 0, \ldots, 0; 0, \ldots, 0; 0, \ldots, 0) = 0$, for any state with $n_1 + \ldots + n_m = n$.

$F(n_1, \ldots, n_m; j_1, \ldots, j_m; k_1, \ldots, k_m; b_1, \ldots, b_m; h_1, \ldots, h_m) = \infty$, for each infeasible state $(n_1, \ldots, n_m;\ j_1, \ldots, j_m;\ k_1, \ldots, k_m;\ b_1, \ldots, b_m; h_1, \ldots, h_m)$. For a state to be feasible, all the following conditions must be satisfied: $n_1 + \ldots + n_m = n$; and for every $i \in M$, $1 \leq j_i \leq \min(k_i, n_i), 0 \leq k_i \leq n,\ h_i \leq b_i,\ b_i \leq \min(b, n_i)$, and the set $Y_{i=1}^{m} SPT_{ik_i}$ contains exactly $j1 + \ldots + j_m$ orders.

Recursive relations

For each feasible state $(n_1, \ldots, n_m; j_1, \ldots, j_m; k_1, \ldots, k_m; b_1, \ldots, b_m; h_1, \ldots, h_m)$:

$F(n_1, \ldots, n_m; j_1, \ldots, j_m; k_1, \ldots, k_m; b_1, \ldots, b_m; h_1, \ldots, h_m) = \min_{\{i \in M\}}$

$$\begin{cases} F(n_1, \ldots, n_m; j_1, \ldots, j_i - 1, \ldots, j_m; k_1, \ldots, k_i', \ldots, k_m; b_1, \ldots, b_m; h_1, \ldots, h_i - 1, \ldots, h_m) \\ \qquad + \alpha[(n_i - j_i + h_i) p_{i[ik_i]} + t_i] + (1-\alpha) c_{i[ik_i]}, \quad \text{if } \ h_i \geq 2 \\ \min_{\{1 \leq b_i \leq b\}} \{F(n_1, \ldots, n_m; j_1, \ldots, j_i - 1, \ldots, j_m; k_1, \ldots, k_i', \ldots, k_m; b_1, \ldots, b_i', \ldots, b_m; \\ \qquad h_1, \ldots, b_i', \ldots, h_m) + \alpha[(n_i - j_i + h_i) p_{i[ik_i]} + t_i] + (1-\alpha)(c_{i[ik_i]} + f_i)\}, \quad \text{if } \ h_i = 1 \end{cases}$$

where k_i' is such that order $[ik_i']$ is the order immediately before order $[ik_i]$ among the remaining orders in the list SPT_i after the orders in the set $Y_{v=1, v \neq i}^{m} SPT_{vk_v}$ have been scheduled.

Optimal solution

An optimal solution is provided by:

minimizing $F(n_1, \ldots, n_m; n_1, \ldots, n_m; k_1, \ldots, k_m; b_1, \ldots, b_m; b_1, \ldots, b_m)$ over all the feasible states $(n_1, \ldots, n_m; n_1, \ldots, n_m; k_1, \ldots, k_m; b_1, \ldots, b_m; b_1, \ldots, b_m)$ with $n_1 + \ldots + n_m = n$.

We note that in this algorithm the partial schedule corresponding to the state $(n_1, \ldots, n_m;$

$j_1, \ldots, j_m;\ k_1, \ldots, k_m;\ b_1, \ldots, b_m;\ h_1, \ldots, h_m)$ contains exactly all the orders in the set $Y_{i=1}^{m} SPT_{ik_i}$. The value function $F(n_1, \ldots, n_m; j_1, \ldots, j_m; k_1, \ldots, k_m; b_1, \ldots, b_m; h_1, \ldots, h_m)$ is obtained by selecting one of the following m alternatives with the lowest total contribution: Appending order $[ik_i]$ to the end of a partial schedule at plant $i \in M$. The partial schedule right before order $[ik_i]$ is added to plant i contains exactly the orders in the set $(Y_{v=1, v\neq i}^{m} SPT_{vk_v}) Y SPT_{ik_i'}$, where k_i' is defined in the algorithm and is unique given $k_1, \ldots, k_m$. There are two possible ways for adding order $[ik_i]$ to plant i:

(1) If the order is added to the current last shipment with final size b_i (in this case, $h_i \geq 2$), then its contribution is $\alpha[(n_i - j_i + h_i)p_{i[ik_i]} + t_i] + (1-\alpha)c_{i[ik_i]}$.

(2) If the order is added to a new shipment with the final size b_i after the current last shipment (in this case, $h_i = 1$) , then its contribution is $\alpha[(n_i - j_i + h_i)p_{i[ik_i]} + t_i] + (1 - \alpha)(c_{i[ik_i]} + f_i)\}$.

Theorem 3 Algorithm DP-P1P solves problem P2 with production costs proportional to processing times to optimality in $O(n^{3m} m b^{2m})$ time.

Proof The algorithm schedules orders at each plant forward from the first position to the last. Lemma 2 shows that given a partial schedule π_L as defined in the statement of the lemma, it is sufficient to consider the following m possible state transitions from a state with j orders to a state with $j+1$ orders in the dynamic program: appending order y_i to the end of π_{iL}, for $i \in M$. In the algorithm, state $(n_1, \ldots, n_m; j_1, \ldots, j_i, \ldots, j_m; k_1, \ldots, k_i, \ldots, k_m)$ can be transitioned from $(n_1, \ldots, n_m; j_1, \ldots, j_i - 1, \ldots, j_m; k_1, \ldots, k_i', \ldots, k_m)$ by adding order $[ik_i]$ to the end of the schedule at plant i, for $i \in M$. All these state transitions are considered in the algorithm, which guarantees the optimality of the algorithm.

There are at most $O(n^{3m} b^m)$ combinations of $(n_1, \ldots, n_m; j_1, \ldots, j_m; k_1, \ldots, k_m, b_1, \ldots, b_m)$ considered in the dynamic program. Let $u(q)$ denote the number of combinations of

$(h_1, \ldots, h_m)$ where the number of h_i's with a value 1 is exactly q $(0 \leq q \leq m)$. Thus, there are no more than $u(q)n^{3m}b^m$ states in the DP where the number of h_i's with a value 1 is exactly q. By a similar argument as in the proof of Theorem 2, it can be shown that the overall time complexity of the algorithm is bounded by $O(n^{3m}mb^{2m})$. ■

Theorem 3 shows that for a fixed number of plants m, the problem P1 with production costs proportional to processing times is solvable in polynomial time. However, due to the high-order time complexity of the algorithm, this algorithm is only of theoretical value and it is impractical to apply it to actually solving the problem. Faster heuristics or approximation algorithms have to be developed to solve the problem.

2.3.4 General Problem P1

Theorem 4 Problem P1 with $0 < \alpha < 1$ and an arbitrary number of plants is strongly NP-hard.

Proof We prove this by a reduction from the strongly NP-hard Minimum Cover (MC) problem, an instance of which is given in the proof of Theorem 1. Given this instance of MC, we consider an instance of the recognition version of problem P1 defined exactly the same way as the one defined in the proof of Theorem 1 except the following parameters:

Order processing times, $p_{ij} = 0$ if $j \in S_i$, and $\lceil \frac{(1-\alpha)}{\alpha} \rceil v + 1$ otherwise, for $i \in M$ and $j \in N$.

Order production costs, $c_{ij} = p_{ij}$ for $i \in M$ and $j \in N$.

Threshold of objective value, $Z = (1 - \alpha)v$.

By similar arguments as in the proof of Theorem 1, we can show that there is a schedule to this instance of P1 with the objective value no more than Z if and only if there is a solution to MC. We omit the details of the proof. ■

The complexity of the problem P1 with $0 < \alpha < 1$ and a fixed number of plants remains open.

In the remainder of this subsection, we propose two heuristics for solving the general problem P1. Before presenting the heuristics, we first derive an upper bound on the size of a shipment containing a given order at each plant. Let $b_{\max,i,j}$ denote the maximum possible size of a shipment containing order j at plant i in an optimal schedule. Denote the SPT order of the orders at plant i by $([i1], [i2], \ldots, [in])$. Suppose that the size of a shipment B containing order j at plant i is x, and B consists of orders $(< i1 >, < i2 >, \ldots, < ix >)$ (where $(< i1 >, < i2 >, \ldots, < ix >)$ is a subset of $([i1], [i2], \ldots, [in])$). Clearly, $p_{i,<iu>} \geq p_{i,[iu]}$, for $u = 1, \ldots, x$. The following computational procedure finds an upper bound on x, and this upper bound is defined as $b_{\max,i,j}$.

Procedure MAXSIZE

Step 0: Initially let $x = b$ (which is the largest possible).

Step 1: Given x, check if the total cost of the orders in shipment B can be reduced by splitting it into two shipments. There are $x-1$ different ways of splitting this shipment into two, E_1 and E_2, with E_1 consisting of the first y orders, i.e. $E_1 = (< i1 >, \ldots, < iy >)$, and E_2 the last $x - y$ orders, i.e. $E_2 = (< i, y + 1 >, \ldots, < ix >)$, for $y = 1, \ldots, x - 1$. For each $y = 1, ..., x - 1$, compute a lower bound of the cost reduction due to the splitting, denoted as R_y as follows:

$$R_y \geq \alpha y(p_{i,[i,y+1]} + p_{i,[i,y+2]} + \ldots + p_{i,[ix-1]} + max\{p_{i,[ix]}, p_{ij}\}) - (1 - \alpha)f_i$$

where $\alpha y(p_{i,[i,y+1]} + p_{i,[i,y+2]} + \ldots + p_{i,[ix-1]} + max\{p_{i,[ix]}, p_{ij}\})$ is a lower bound of the decrease in total lead time of the orders in E_1 and $(1 - \alpha)f_i$ is the increase in delivery cost.

Step 2: Find $z \in \{1, \ldots, x - 1\}$ such that $R_z = \max\{R_y | y = 1, \ldots, x - 1\}$. If $R_z \geq 0$, then the shipment size x should be reduced; update $x = x - 1$ and go to Step 1. If $R_z < 0$, then stop, and the current x is the maximum size of a shipment containing order j at plant i.

We first give a base heuristic for solving the general problem P1. Subsequently, we will propose an improved heuristic for P1 based on this base heuristic. The heuristics also works for the problem P1 with $\alpha = 0$ (which is proved in Theorem 1 to be NP-hard if the number of plants m is arbitrary). The base heuristic consists of an initialization step, where some parameters used in later steps are calculated, and a two-phase procedure, where the orders are assigned to the plants in the first phase, and then the orders assigned to each plant are scheduled in the second phase. The order assignment problem in the first phase is solved as a standard assignment problem where the cost of assigning an order to a particular position of a plant is heuristically derived based on relevant parameters of the order and the parameters calculated in the initialization step. The order scheduling problem in the second phase is solved by a dynamic programming algorithm.

Heuristic H1-BASE

Initialization: Run procedure MAXSIZE to derive the maximum shipment sizes $b_{\max,i,j}$ for $i \in M$ and $j \in N$. Define parameters D_{ih} and e_{ui}, for $i \in M, h = 1, \ldots, n-1$, and $u = 1, \ldots, b$, as follows:

Δ_{ih} = the minimum difference of processing times of two orders that are h orders apart in the SPT order of the orders at plant i. That is, if the SPT order of the orders at plant i is $([i1], [i2], \ldots, [in])$, then: $\Delta_{ih} = \min\{p_{i,[i,q+h]} - p_{i,[iq]} | q = 1, 2, \ldots, n-h\}$ (which is always nonnegative).

$$e_{ui} = \begin{cases} \frac{1}{2u}[\Delta_{i1} + 3\Delta_{i3} + 5\Delta_{i5} + \ldots + (u-1)\Delta_{i,u-1}], \text{if } u \text{ is even} \\ \\ \frac{1}{2u}[2\Delta_{i2} + 4\Delta_{i4} + 6\Delta_{i6} + \ldots + (u-1)\Delta_{i,u-1}], \text{if } u \text{ is odd} \end{cases}$$

Phase 1: For $k = 1, \ldots, n, i = 1, \ldots, m$, and $j = 1, \ldots, n$, define parameter

$$a_{(k,i)j} = \alpha(kp_{ij} + t_i) + (1-\alpha)c_{ij} + \min_{1 \le u \le b_{\max,i,j}} \{\alpha p_{ij}(u-1)/2 + (1-\alpha)f_i/u + \alpha e_{ui}\} \quad (2.1)$$

Define a binary variable $x_{(k,i)j}$ to be 1 if order j is scheduled as the kth last order at plant i, and 0 otherwise. Solve the following assignment problem. Let the optimal solution be denoted as π_1.

$$\min \quad G = \sum_{k \in N} \sum_{i \in M} \sum_{j \in N} a_{(k,i)j} x_{(k,i)j}$$

Subject to:

$$\begin{aligned} \sum_{k \in N} \sum_{i \in M} x_{(k,i)j} &= 1 \quad j \in N \\ \sum_{j \in N} x_{(k,i)j} &\le 1 \quad k \in N \quad i \in M \\ x_{(k,i)j} \in \{0,1\} & \qquad\quad k \in N, i \in M, j \in N \end{aligned}$$

Phase 2: Given π_1, for the orders scheduled at each plant, find an optimal delivery schedule with respect to the objective function of P1. Let the solution be denoted as π_2.

We note that the cost coefficients $a_{(k,i)j}$ in the assignment problem formulated in Phase 1 of the heuristic are defined in (2.1) based on the following observation. If order j is scheduled as the kth last order at plant i, then the contribution of order j to the objective value is exactly $\alpha((k+r)p_{ij} + t_i) + (1-\alpha)(c_{ij} + f_i/q)$, where q is the number of orders in the shipment containing order j and r is the number of orders scheduled before order j in this shipment. Since we do not know the values of q and r, the term involving q and r, i.e.,

$\alpha r p_{ij} + (1-\alpha) f_i/q$, is approximated by $\min_{\{1 \le u \le b_{\max,i,j}\}} \{\alpha p_{ij}(u-1)/2 + (1-\alpha) f_i/u + \alpha e_{ui}\}$. It can be shown that the orders are scheduled in the SPT order at each plant in the solution π_1 generated in Phase 1. In Phase 2, the problem of finding an optimal delivery schedule at each plant $i \in M$ given the order processing schedule π_1 can be solved in polynomial time by the following dynamic programming algorithm. Suppose that there are n_i orders processed by plant i and their SPT order is $([i1], \ldots, [in_i])$, where the notation $[ih]$ represents the index of the hth order. Let $C_{[ij]}$ denote the time when order $[ij]$ completes processing at plant i under schedule π_1.

Algorithm DP-PHASE2

Define value function $V(j)$ = minimum total cost of a schedule for the first j orders $\{[i1], \ldots, [ij]\}$.

Initial condition: $V(0) = 0$.

Recursive relation: For $j = 1, \ldots, n_i$,

$$V(j) = min\{V(j-h) + \alpha h(C_{[ij]} + t_i) + (1-\alpha) f_i | h = 1, ..., \min(b, j)\}$$

Optimal solution: $V(n_i)$.

The optimality of algorithm DP-PHASE2 follows from the fact that the recursive relation tries every possible size h of the last delivery shipment. This algorithm has a time complexity $O(n_i b)$.

For ease of presentation, in the remainder of this subsection, we denote the objective function of problem P1 as F (i.e. $F = \alpha D_{\text{total}} + (1-\alpha)TC$), that of a particular schedule π as $F(\pi)$, and that of an optimal schedule as F^*. Similarly, we denote the objective function of the assignment problem in Phase 1 of the heuristic as G and that of a particular solution

π as $G(\pi)$.

Lemma 4 $F^* \geq G(\pi_1)$, where π_1 is the solution generated in Phase 1 of the heuristic H1-BASE.

Proof Given an optimal solution π^* of P1 that satisfies Lemma 1, consider any shipment of orders, denoted as B, at any plant i. Suppose that there are $h(1 \leq h \leq min\{b_{\max,i,j} | j \in B\})$ orders in B which are indexed as $([h], [h-1], \ldots, [1])$, and that order [1] is scheduled at the kth last position at plant i, for some $k \geq 1$. Since π^* satisfies Lemma 1, the orders are scheduled in the SPT order at each plant. This means that

$$p_{i[1]} \geq p_{i[2]} \geq \ldots \geq p_{i[h]} \tag{2.2}$$

The total contribution of the orders in B to the objective function F under schedule π^*, denoted as $C(B)$, is

$$\begin{aligned} C(B) &= \sum_{j=1}^{h} \left[\alpha(k+h-1)p_{i[j]} + \alpha t_i + (1-\alpha)c_{i[j]}\right] + (1-\alpha)f_i \\ &= \sum_{j=1}^{h} \left[\alpha(k+j-1)p_{i[j]} + \alpha(h-j)p_{i[j]} + \alpha t_i + (1-\alpha)c_{i[j]} + (1-\alpha)f_i/h\right] \end{aligned} \tag{2.3}$$

In the following we derive a lower bound of $C(B)$. First we evaluate the summation $\sum_{j=1}^{h}(h-j)p_{i[j]}$. Consider two cases of h:

Case 1: if h is even, i.e. $h = 2g$, for some integer $g \geq 1$, then by (2.2),

$$\begin{aligned} \sum_{j=1}^{h}(h-j)p_{i[j]} &= \sum_{j=1}^{g}\left[(h-j)p_{i[j]} + (j-1)p_{i[h-j+1]}\right] \\ &= \sum_{j=1}^{g}\left[\frac{(h-1)}{2}p_{i[j]} + \frac{(h-1)}{2}p_{i[h-j+1]} + \frac{h-2j+1}{2}(p_{i[j]} - p_{i[h-j+1]})\right] \\ &\geq \sum_{j=1}^{g}\left[\frac{(h-1)}{2}p_{i[j]} + \frac{(h-1)}{2}p_{i[h-j+1]} + \frac{h-2j+1}{2}\Delta_{i,h-2j+1}\right] \\ &= \sum_{j=1}^{h}\left[\frac{(h-1)}{2}p_{i[j]}\right] + he_{hi} = \sum_{j=1}^{h}\left[\frac{(h-1)}{2}p_{i[j]} + e_{hi}\right] \end{aligned}$$

Case 2: if h is odd, i.e. $h = 2g+1$, for some integer $g \geq 0$, then by (2.2),

$$\begin{aligned}
\sum_{j=1}^{h}(h-j)p_{i[j]} &= \sum_{j=1}^{g}\left[(h-j)p_{i[j]} + (j-1)p_{i[h-j+1]}\right] + gp_{i[g+1]} \\
&= \sum_{j=1}^{g}\left[\frac{(h-1)}{2}p_{i[j]} + \frac{(h-1)}{2}p_{i[h-j+1]} + \frac{h-2j+1}{2}(p_{i[j]} - p_{i[h-j+1]})\right] \\
&\qquad + gp_{i[g+1]} \\
&\geq \sum_{j=1}^{g}\left[\frac{(h-1)}{2}p_{i[j]} + \frac{(h-1)}{2}p_{i[h-j+1]} + \frac{h-2j+1}{2}\Delta_{i,h-2j+1}\right] + gp_{i[g+1]} \\
&= \sum_{j=1}^{h}\left[\frac{(h-1)}{2}p_{i[j]}\right] + he_{hi} = \sum_{j=1}^{h}\left[\frac{(h-1)}{2}p_{i[j]} + e_{hi}\right]
\end{aligned}$$

Combining the above two cases of h, we have

$$\sum_{j=1}^{h}(h-j)p_{i[j]} \;\geq\; \sum_{j=1}^{h}\left[\frac{(h-1)}{2}p_{i[j]} + e_{hi}\right] \tag{2.4}$$

By (2.3) and (2.4), the total contribution of the orders in shipment B to the objective function F under schedule π^* satisfies the following:

$$\begin{aligned}
C(B) &\geq \sum_{j=1}^{h}\left[\alpha(k+j-1)p_{i[j]} + \alpha\frac{(h-1)}{2}p_{i[j]} + \alpha e_{hi} + \alpha t_i + (1-\alpha)c_{i[j]} + (1-\alpha)\frac{f_i}{h}\right] \\
&= \sum_{j=1}^{h}\left[\alpha\left((k+j-1)p_{i[j]} + t_i\right) + (1-\alpha)c_{i[j]} + \alpha\frac{(h-1)}{2}p_{i[j]} + (1-\alpha)\frac{f_i}{h} + \alpha e_{hi}\right] \\
&\geq \sum_{j=1}^{h} a_{(k+j-1,i)[j]} \quad \text{by (2.1)}
\end{aligned}$$

This means that under schedule π^*, the total contribution of the orders in B to F is greater than or equal to the total contribution of the same orders to G if they are scheduled at the same positions. Since this is true for every shipment at every plant, we have: $F^* = F(\pi^*) \geq G(\pi^*)$. Since π_1 is optimal with respect to G, $G(\pi^*) \geq G(\pi_1)$. Therefore, $F^* \geq G(\pi_1)$. ■

Lemma 4 shows that the optimal objective value of the assignment problem in Phase 1 of the heuristic is a lower bound of the optimal objective value of problem P1. The following theorem gives the worst-case performance of the heuristic.

Theorem 5 Let π_2 be the schedule generated by heuristic H1-BASE. Then $F(\pi_2) \leq b_{\max} F^*$, where $b_{\max} = \max\{b_{\max,i,j} | i \in M, j \in N\}$. In other words, the worst-case performance ratio of heuristic H1-BASE for problem P1 is bounded by $b_{\max}$. Furthermore, this bound is tight.

Proof Based on solution π_1 obtained in Phase 1 of the heuristic, we construct a solution Γ to problem P1 as follows. Let the order assignment and order sequence at each plant be exactly the same as in π_1, and deliver each order separately immediately after it completes processing. Then we have the following two results:

(i) $F(\Gamma) \geq F(\pi_2)$ because in π_2 orders are delivered optimally given the order assignment and sequences as specified by π_1.

(ii) In Γ, if order j is scheduled at the kth last position at plant i, its contribution to the objective function F is $\alpha(kp_{ij} + t_i) + (1 - \alpha)c_{ij} + (1 - \alpha)f_i \leq b_{\max,i,j} a_{(k,i)j}$, where $a_{(k,i)j}$ is defined in (2.1).

Result (ii) implies that $F(\Gamma) \leq b_{\max} G(\pi_1)$. By Lemma 4 and Result (i), we have $F(\pi_2) \leq b_{\max} F^*$.

To show that this bound is tight, consider the following instance of the problem: there are b orders $N = \{1, \ldots, b\}$ and b plants $M = \{1, \ldots, b\}$ with $p_{ij} = c_{ij} = 0$, $t_i = 0$, and $f_i = f$, for $i \in M$ and $j \in N$, where f is any positive constant. For this example, applying procedure MAXSIZE does not reduce the maximum size of a shipment because splitting a shipment into two does not reduce the total cost. Therefore, $b_{\max,i,j} = b$ for $i \in M$ and $j \in N$. In Phase 1 of the heuristic, $a_{(k,i)j} = (1 - \alpha)f/b$ for $k, j \in N$ and $i \in M$. One of the optimal solutions of the assignment problem in Phase 1 is to assign one order to each plant. If this solution is used to generate the final solution in Phase 2, each order will be delivered in a separate shipment. The objective value of this solution is $(1 - \alpha)bf$. On the other hand, an optimal solution to problem P1 is to schedule all the orders at a single plant and deliver them by one shipment, which yields the optimal objective value $(1-\alpha)f$. Thus the ratio of the objective value generated by the heuristic and the optimal objective value is b. ■

Theorem 3 means that the worst-case performance ratio of H1-BASE is input data dependent and may be large when b is large. However, the next theorem shows that the heuristic is capable of generating near-optimal solutions for problems with a large number of orders.

Theorem 6 If all the order processing times p_{ij} in problem P1 are nonzero and finite, then the solution π_2 generated by H1-BASE is asymptotically optimal for P1 when n goes to infinity with m and b fixed.

Proof There are two cases to consider: (i) $\alpha = 0$; (ii) $\alpha > 0$.

In Case (i), the problem is to minimize total production and transportation cost. In this case, in an optimal schedule, all the delivery shipments are full (i.e. with a size b) except possibly one at each plant. Thus $b_{\max,i,j} = b$, for $i \in M$ and $j \in N$. By (2.1), $a_{(k,i)j} = c_{ij} + f_i/b$. This means that the assignment problem in Phase 1 of the heuristic assumes that the contribution to the transportation cost by each order assigned to plant i is f_i/b, which underestimates the true contribution if it is delivered in a shipment with a size less than b. In schedule π_2 generated in Phase 2, all the delivery shipments are full except possibly one shipment at each plant. Therefore, there are at most $b-1$ orders at each plant whose contribution to the transportation cost is underestimated by the assignment problem in Phase 1. This implies that $F(\pi_2) \leq G(\pi_1) + (f_1 + \ldots + f_m)$. By Lemma 2.3 and the fact that $F^* \leq F(\pi_2)$, we have

$$F^* \leq F(\pi_2) \leq F^* + (f_1 + \ldots + f_m) \tag{2.5}$$

When $n \to \infty$, with m fixed, F^* dominates $f_1 + \ldots + f_m$, i.e. $\lim_{n\to\infty} \frac{f_1+\ldots+f_m}{F^*} = 0$. By (2.5), we have

$$\lim_{n\to\infty} \frac{F(\pi_2) - F^*}{F^*} = 0$$

In Case (ii), we construct a solution Γ to problem P1 based on solution π_1 generated in Phase 1 as follows. Let the order assignment and order sequence at each plant be exactly the same as in π_1, and deliver each order in a separate shipment immediately after it completes processing. Since orders in π_2 are delivered optimally given π_1, we have $F(\Gamma) \geq F(\pi_2)$. Hence, by Lemma 4,

$$G(\pi_1) \leq F^* \leq F(\pi_2) \leq F(\Gamma) \tag{2.6}$$

In Γ, if order j is scheduled at the kth last position at some plant i, its contribution to the objective function F is $q_{(k,i)j} = \alpha(kp_{ij} + t_i) + (1-\alpha)c_{ij} + (1-\alpha)f_i$. Let $F_1(\Gamma)$ and $F_2(\Gamma)$ denote the part of $F(\Gamma)$ contributed by the cost terms $\alpha(kp_{ij} + t_i) + (1-\alpha)c_{ij}$ and $(1-\alpha)f_i$, respectively. Also, let $G_1(\pi_1)$ and $G_2(\pi_1)$ denote the part of $G(\pi)$ contributed by the cost terms $\alpha(kp_{ij} + t_i) + (1-\alpha)c_{ij}$ and $\min_{\{1\leq u\leq b_{\max,i}\}}\{\alpha p_{ij}(u-1)/2 + (1-\alpha)f_i/u + \alpha e_{ui}\}$, respectively. Since Γ has the same order assignment and sequence at each plant as π_1, we have

$$F_1(\Gamma) = G_1(\pi_1) \tag{2.7}$$

Since all the order processing times p_{ij} are nonzero and finite, there exists a positive integer L such that $\alpha kp_{ij} \geq \alpha kL > (1-\alpha)f_i$ for k greater than a certain value, and the gap between αkp_{ij} and $(1-\alpha)f_i$ grows linearly with k. As $n \to \infty$, $F_1(\Gamma)$ dominates $F_2(\Gamma)$, i.e. $\lim_{n\to\infty} \frac{F_2(\Gamma)}{F_1(\Gamma)} = 0$. Similarly, as $n \to \infty$, $G_1(\pi_1)$ dominates $G_2(\pi_1)$, i.e. $\lim_{n\to\infty} \frac{G_2(\pi_1)}{G_1(\pi_1)} = 0$. This, along with (2.7), implies that

$$\lim_{n\to\infty} \frac{F(\Gamma) - G(\pi_1)}{G(\pi_1)} = \lim_{n\to\infty} \frac{F_2(\Gamma) - G_2(\pi_1)}{F_1(\Gamma) + G_2(\pi_1)} = 0$$

By (2.6), this implies that

$$0 \leq \lim_{n \to \infty} \frac{F(\pi_2) - F^*}{F^*} \leq \lim_{n \to \infty} \frac{F(\Gamma) - G(\pi_1)}{G(\pi_1)} = 0$$

This means that $\lim_{n\to\infty} \frac{F(\pi_2)-F^*}{F^*} = 0$ i.e., π_2 is asymptotically optimal for P1 as n goes to infinity. ■

Our computational experiment (described later) shows that heuristic H1-BASE is capable of generating near optimal solutions for most of our test problems. However, for a small subset of the test problems with a relatively large b and small n, the performance is not satisfactory. Since the heuristic we propose in the next section for problem P2 builds on this heuristic, it is worthwhile to improve the performance of this heuristic as much as possible. We therefore propose another heuristic which tries to improve the solution generated by H1-BASE by lowering the maximum shipment size $b_{\max,i,j}$ for each order j at each plant i and rerunning H1-BASE with the revised $b_{\max,i,j}$.

Heuristic H1-IMP

Step 1: Revise the maximum batch size $b_{\max,i,j}$ for each order $j \in N$ at each plant $i \in M$ based on the solution π_2 generated by H1-BASE as follows. Suppose there are q_i shipments of orders scheduled at plant $i \in M$ in the solution π_2. Let $B_{i1}, \ldots, B_{iq_i}$ be those shipments scheduled in this order. For each order $j \in N$ and each plant $i \in M$, if p_{ij} is between the processing times of two orders within some shipment B_{ir}, then redefine $b_{\max,i,j}$ to be equal to $|B_{ir}|$. If p_{ij} is between the processing times of two orders which are in two separate shipments B_{ir} and $B_{i,r+1}$, then redefine $b_{\max,i,j}$ to be equal to $|B_{i,r+1}|$.

Step 2: Re-run Heuristic H1-BASE with the revised $b_{\max,i,j}$. Let the solution generated in Phase 2 be π'_2. Choose the better one of π_2 (generated by the original heuristic H1-BASE) and π_2' as the solution for problem P1. Denote this solution as π_2^{IMP}.

We note that with the revised shipment sizes $b_{\max,i,j}$, the solution generated in Phase 1, denoted as π_1', may not satisfy Lemma 4, i.e. $G(\pi_1')$ is not necessarily a lower bound of F^*. This is because the revised $b_{\max,i,j}$ may not be a valid upper bound of the size of a batch containing order j at plant i. However, the solution π_2^{IMP} generated by H1-IMP satisfies Theorems 5 and 6 because it is always no worse than π_2.

Next we conduct a computational experiment to evaluate the performance of H1-BASE and H1-IMP based on random test problems generated as follows.

a. Number of orders $n \in \{50, 100, 200\}$; number of plants $m \in \{2, 4, 8\}$; shipment capacity $b \in \{3, 6, 12\}$

b. Order processing times p_{ij} are independently generated from a uniform distribution $U[10, 100]$

c. Two types of order production costs c_{ij} are considered. Type 1: c_{ij} are independently generated from a uniform distribution $U[10, 500]$; Type 2: c_{ij} are proportional to production times, i.e. $c_{ij} = \gamma_i p_{ij}$ where γ_i are independently generated from a uniform distribution $U[1, 10]$

d. Transportation times t_i are independently generated from a uniform distribution $U[100, 1000]$; transportation costs per delivery shipment f_i is proportional to the delivery times, i.e. $f_i = \rho_i t_i$, where $\rho_i \in \{0.5, 1, 2\}$

e. Weighting parameter in the objective function $\alpha \in \{0.2, 0.5, 0.8\}$.

We note that test problems generated this way represent a wide variety of practical situations as follows: (i) The delivery time t_i varies from 1 to 100 times an order processing time p_{ij} ; (ii) The average delivery cost per shipment f_i (about $550\rho_i$) varies from 1 to 4 times the average order processing cost c_{ij} (about 250) when ρ_i varies from 0.5 to 2; and (iii) The weighting parameter α covers a wide range of the interval $[0, 1]$. In practice, it is often the case that the production cost of an order at a plant is proportional to the processing time of the order. This is reflected by the use of the Type-2 scheme for generating c_{ij}.

For each of the 243 combinations of the five parameters with multiple choices $(n, m, b, \rho_i, \alpha)$, we test 20 randomly generated instances, 10 with c_{ij} generated following the Type-1 scheme, and 10 with c_{ij} generated following the Type-2 scheme. Every test problem is solved in no more than 10 CPU seconds. (Note that all the heuristics in this chapter are coded in C++ and run on a PC with a 1.5-GHz Pentium IV processor and 512-MB memory. All the LP problems, including the assignment problem in Phase 1 of H1-BASE, are solved by calling the LP Solver of CPLEX, Version 8). Table 2.1 reports both average and maximum relative gaps between the objective values of the solutions generated by H1-BASE and H1-IMP and the lower bound $G(\pi_1)$. For a test problem, the relative gap between the objective value of the solution generated by H1-BASE and $G(\pi_1)$ is defined as $\frac{F(\pi_2)-G(\pi_1)}{G(\pi_1)} \times 100\%$. The relative gap between the objective value of the solution generated by H1-IMP and $G(\pi_1)$ is defined similarly. Clearly, the relative gaps defined here are upper bounds of the actual relative gaps between the heuristic solutions and the optimal solution. Each entry in the columns "Avg Gap" ("Max Gap") of Table 2.1 is the average (maximum) relative gap over the 180 random test problems with the corresponding (n, m, b) combination, 20 for each of the nine (ρ_i, α) combinations. The number in the parentheses next to each maximum relative gap over 10% is the number of test problems (out of 180 problems) for which the relative gap is at least 10%.

These results demonstrate that both heuristics are capable of generating near optimal solutions for most problems tested. The average relative gap of H1-BASE over all the 4860 test problems is 1.36%, whereas that of H1-IMP is 1.05%, a more than 20% improvement from H1-BASE. Figures 2.2 and 2.3 show the asymptotic optimality property of H1-IMP with respect to the number of orders. There are a total of 100 test problems (about 2% of all the test problems used) for which H1-BASE generates a solution with a 10% or more relative gap, whereas this number is 35 (about 0.7%) in the case of H1-IMP. It can also be observed that the relative gap generally decreases with n and increases with m and b.

2.4 Problem P2: Minimizing TC subject to $D_{\text{total}} \leq D$

We first show that the problem is at least ordinarily NP-hard even with two plants $m = 2$ and under the two special cases noted in the beginning of Section 2.3: (i) order processing times are agreeable; (ii) production costs are proportional to processing times. Then we propose a heuristic for the general problem P2 and evaluate its performance computationally.

Theorem 7 Problem P2 is at least ordinarily NP-hard even when there are only two plants and both of the two special cases (i) and (ii) hold.

Proof We prove this by a reduction from the known ordinarily NP-hard Equal-Size Partition Problem (ESPP) (Garey and Johnson 1979): ESPP:Given $2h$ items $H = \{1, \ldots, 2h\}$. Each item $i \in H$ has a known integer size a_i, such that $\sum_{i \in H} a_i = 2A$, for some integer A. The question asks: does there exist a subset of the items $Q \subseteq H$ such that it contains exactly h items and the total size $\sum_{j \in Q} a_j = A$?

Given an instance of ESPP, we create the following instance for the recognition version of our problem:

Number of orders $n = 2h$, and order set $N = H$

Number of plants $m = 2$

Shipment capacity $b = h$

Transportation cost of a shipment $f_1 = f_2 = M$ sufficiently large

Transportation time $t_1 = t_2 = 0$

Processing time of order $j \in N$: $p_{1j} = Aa_j$, $p_{2j} = a_j$

Production cost of order $j \in N$: $c_{1j} = a_j$, $c_{2j} = Aa_j$

Upper bound on total delivery time $D = hA^2 + hA$

Threshold on total cost $F = 2M + A^2 + A$.

We prove that there is a solution to our problem with the total production and transportation cost no more than F and total delivery time no more than D if and only if there exists a solution to ESPP.

(If part) Given a subset $Q \subseteq H$ for ESPP with $\sum_{j \in Q} a_j = A$ and $|Q| = h$, we construct a solution to our problem as follows. Process orders from Q at plant 1 and deliver them in one shipment at time A^2. Process the rest of the orders at plant 2 and deliver them in one shipment at time A. In this schedule, $D_{\text{total}} = \sum_{j \in Q} Dj + \sum_{j \in H \setminus Q} Dj = h \sum_{j \in Q} Aa_j + h \sum_{j \in H \setminus Q} a_j = hA^2 + hA = D$, the total production cost is $\sum_{j \in Q} c_{1j} + \sum_{j \in H \setminus Q} c_{2j} = \sum_{j \in Q} a_j + A \sum_{j \in H \setminus Q} a_j = A + A^2$, and the total transportation cost is $2M$. Therefore the total cost is exactly F.

(Only if part) Given a solution to our problem with the total cost no more than F and total delivery time of the orders no more than D, we can conclude that there must be at most two shipments because otherwise the total cost would be more than $3M > F$. Since $b = h$, in this solution, there must be exactly two shipments, each containing exactly h orders. Consider two cases:

Case 1: If the orders of both shipments are processed at plant 1, then D_{total} will be more than $(\sum_{j\in H} p_{1j})h = (\sum_{j\in H} Aa_j)h = 2hA^2 > D$. This violates the constraint that D_{total} is no more than D.

Case 2: If the orders of both shipments are processed at plant 2, then the total cost will be more than $2f_2 + \sum_{j\in H} c_{2j} = 2M + \sum_{j\in H} Aa_j = 2M + 2A^2 > F$, which is in contradiction with the fact that the total cost of the schedule is no more than F.

Hence each plant processes the orders of one shipment. Let R denote the set of orders processed at plant 1. Then the total delivery time of orders is

$$D_{\text{total}} = \left(\sum_{j\in R} p_{1j}\right) h + \left(\sum_{j\in N\setminus R} p_{2j}\right) h = hA\sum_{j\in R} a_j + h\left(2A - \sum_{j\in R} a_j\right) \leq D$$

which means that $\sum_{j\in R} a_j \leq A$. The total cost is

$$TC = f_1 + f_2 + \sum_{j\in R} c_{1j} + \sum_{j\in N\setminus R} c_{2j} = 2M + \sum_{j\in R} a_j + A\left(2A - \sum_{j\in R} a_j\right) \leq F$$

which means that $\sum_{j\in R} a_j \geq A$. Therefore, $\sum_{j\in R} a_j = A$ and subset R is a solution to ESPP.

In the constructed instance of our problem, there are only two plants and both of the special cases (i) and (ii) hold. Hence we can conclude that problem P2 with two plants and under the two special cases is at least ordinarily NP-hard. ■

We note that problem P1 with either of the special cases (i), (ii) and a fixed number of plants is solvable in polynomial time. Theorem 7 means that problem P2 is more difficult than problem P1 at least for the case when m is fixed and processing times are agreeable or production costs are proportional to processing times.

2.4.1 A Heuristic for Problem P2

The logic of the heuristic is based on the following observation regarding problem P1.

Lemma 5 The total delivery time of orders D_{total} in an optimal solution of P1 is non-increasing with the weighting parameter α in the objective function of the problem, whereas the total cost TC in an optimal solution of P1 is non-decreasing with α.

Proof We prove the first part of the result by contradiction. The second part can be proved similarly. Suppose that there exists $\alpha_1, \alpha_2 \in [0, 1]$ with $\alpha_2 > \alpha_1$ such that the total delivery time in the optimal solution of P1 with $\alpha = \alpha_2$ is greater than that in the optimal solution of P1 with $\alpha = \alpha_1$. Let ρ_i be the optimal solution of P1 with $\alpha = \alpha_i$ for $i = 1, 2$. Denote the total delivery time and total cost in a solution ρ by $D_{\text{total}}(\rho)$ and $TC(\rho)$, respectively. We have

$$\alpha_1 D_{\text{total}}(\rho_1) + (1 - \alpha_1)TC(\rho_1) \leq \alpha_1 D_{\text{total}}(\rho_2) + (1 - \alpha_1)TC(\rho_2) \tag{2.8}$$

$$\alpha_2 D_{\text{total}}(\rho_2) + (1 - \alpha_2)TC(\rho_2) \leq \alpha_2 D_{\text{total}}(\rho_1) + (1 - \alpha_2)TC(\rho_1) \tag{2.9}$$

By (2.9), we have

$$TC(\rho_1) \geq \alpha_2[D_{\text{total}}(\rho_2) - D_{\text{total}}(\rho_1)]/(1 - \alpha_2) + TC(\rho_2)$$

which implies that

$$\begin{aligned}
&\alpha_1 D_{\text{total}}(\rho_1) + (1 - \alpha_1)TC(\rho_1) \\
&\geq \alpha_1 D_{\text{total}}(\rho_1) + (1 - \alpha_1)\alpha_2[D_{\text{total}}(\rho_2) - D_{\text{total}}(\rho_1)]/(1 - \alpha_2) + (1 - \alpha_1)TC(\rho_2) \\
&\geq \alpha_1 D_{\text{total}}(\rho_1) + \alpha_2[D_{\text{total}}(\rho_2) - D_{\text{total}}(\rho_1)] + (1 - \alpha_1)TC(\rho_2) \quad \text{(since } \alpha_2 > \alpha_1\text{)} \\
&= \alpha_1 D_{\text{total}}(\rho_2) + (1 - \alpha_1)TC(\rho_2) + (\alpha_2 - \alpha_1)[D_{\text{total}}(\rho_2) - D_{\text{total}}(\rho_1)] \\
&> \alpha_1 D_{\text{total}}(\rho_2) + (1 - \alpha_1)TC(\rho_2) \\
&\qquad\qquad\qquad \text{(since } \alpha_2 > \alpha_1 \text{ and by assumption } D_{\text{total}}(\rho_2) > D_{\text{total}}(\rho_1)\text{)}
\end{aligned}$$

which is in contradiction with (2.8). This shows that the assumption that $D_{\text{total}}(\rho_2) > D_{\text{total}}(\rho_1)$ cannot hold. ■

Lemma 5 means that problem P2 is equivalent to the problem of finding a minimum $\alpha_0 \in [0,1]$ such that the total delivery time of the orders in an optimal solution of problem P1 with $\alpha = \alpha_0$ is no more than D. Based on this observation, we propose a heuristic to solve P2 by solving P1 multiple times, each time with a different α in the objective function. The framework of the heuristic essentially follows the well-known line search algorithm in the nonlinear programming literature (e.g. Bazaraa et al. 1993). It searches for a minimum possible $\alpha_0 \in [0,1]$ such that when P1 with $\alpha = \alpha_0$ is solved by Heuristic H1-IMP of Section 2.3, the total delivery time of the orders in the solution is no more than D.

Heuristic H2

Step 0: Set $\alpha_0 = 0$. Apply Heuristic H1-IMP to problem P1 with $\alpha = \alpha_0$. If the solution is feasible to problem P2 (i.e. if $D_{\text{total}} \leq D$ in this solution), then stop. This solution is optimal to P2 and it is adopted. Otherwise, set $\delta = 0.5$, and $\alpha_0 = 0.5$.

Step 1: Apply Heuristic H1-IMP to problem P1 with $\alpha = \alpha_0$. If the solution is feasible to problem P2, set $\delta = \delta/2$ and $\alpha_0 = \alpha_0 - \delta$. Otherwise, set $\delta = \delta/2$ and $\alpha_0 = \alpha_0 + \delta$. Repeat Step 1 until δ reaches a prespecified error tolerance (e.g. 0.01). Adopt the last feasible solution to P2.

Along with a feasible solution to problem P2, a lower bound of the optimal objective value of P2 is obtained by Heuristic H2 as follows. Let π^* denote an optimal solution of P2. For every α_0 tried in the heuristic, let $LB(\alpha_0)$ denote the optimal objective value of the assignment problem in Phase 1 of Heuristic H1-BASE. By Lemma 4, $LB(\alpha_0)$ is a lower bound of the optimal objective value of problem P1 with $\alpha = \alpha_0$. Thus, $LB(\alpha_0)$ is also a lower bound of the objective value of solution π^* for P1 with $\alpha = \alpha_0$, i.e.

$$LB(\alpha_0) \leq \alpha_0 D_{\text{total}}(\pi^*) + (1 - \alpha_0)TC(\pi^*)$$

which implies that

$$[LB(\alpha_0) - \alpha_0 D_{\text{total}}(\pi^*)]/(1 - \alpha_0) \leq TC(\pi^*)$$

Since $D_{\text{total}}(\pi^*) \leq D$, the above inequality further means that

$$[LB(\alpha_0) - \alpha_0 D]/(1 - \alpha_0) \leq TC(\pi^*)$$

Since this is true for every $\alpha_0 \in [0, 1]$, we can take the maximum of the left-hand side of the above inequality over all the α_0 values that are tried in H2 as a lower bound of $TC(\pi^*)$, i.e.

$$TC(\pi^*) \geq \max\{[LB(\alpha_0) - \alpha_0 D]/(1 - \alpha_0)| \text{ all } \alpha_0 \text{ tried in H2 }\} \tag{2.10}$$

Now we computationally evaluate the performance of H2. All the parameters except D are generated exactly the same way as in the test for H1-BASE and H1-IMP in Section 2.3. Parameter D is given by $D = D^1 + \beta(D^0 - D^1)$ where $\beta \in \{0.25, 0.5, 0.75\}$, and D^1 and D^0 are minimum and maximum possible values of total delivery time of orders in a schedule, respectively. The value of D^1 is equal to the total delivery time of the orders in the solution of problem P1 with $\alpha = 1$ obtained by solving the assignment problem formulated in Section 2.3. The value of D^0 is equal to the total delivery time of the orders in the solution of P1 with $\alpha = 0$ obtained by H1-IMP.

Similar to the computational experiment in Section 2.3, for each of the 243 combinations of the five parameters with multiple choices (n, m, b, ρ_i, β), we test 20 randomly generated instances, 10 with c_{ij} generated following the Type-1 scheme, and 10 with c_{ij} generated following the Type-2 scheme. Table 2.2 reports the computational results, where each column except the last one represents the same performance measure as the corresponding on in Table 2.1. The relative gap of a test problem is the relative gap between the objective value of the solution generated by H2 and the lower bound given on the right-hand side of (10). Since the computational times required by large problems are not negligible (e.g. within 10 seconds), we also report the average CPU times in seconds in the column "CPU (seconds)". The variance of CPU times over the 180 test problems for each (n, m, b) combination is very small. Hence we do not report the maximum CPU times.

It can be seen from the results in Table 2.2 that (i) the average gap over all the problems with 50, 100, and 200 orders are $8.54\%, 4.93\%$, and 3.08%, respectively; and (ii) the percent of test problems with 50, 100, and 200 orders that have a relative gap over 10% are $29.8\%, 12.0\%$, and 3.7%, respectively. These results show that the performance of H2 improves with the number of orders n in a test problem. For fixed n, the heuristic performs better for problems with smaller m and b. Overall it is capable of generating near-optimal solutions for most test problems with 100 orders or more. Although we are unable to prove it, we conjecture that H2 is asymptotically optimal for problem P2 when the number of orders n goes to infinity with m and b fixed.

2.5 Problem P3: Minimizing $\alpha D_{\max} + (1 - \alpha)TC$

When $\alpha = 0$, this problem reduces to the problem of minimizing total production and distribution cost TC, which has been discussed in Section 2.3. When $\alpha \in (0, 1]$, the problem is at least ordinarily NP-hard even with a fixed number of plants and agreeable processing times. This is because the following special case of the problem: $m = 2, p_{ij} = p_j, c_{ij} = 0, t_i = 0, f_i = 0$ for all $i \in M$ and $j \in N$, is equivalent to the parallel machine maximum completion time scheduling problem, a known ordinarily NP-hard problem (Garey and

Johnson 1979). The following result holds for problem P3.

Lemma 6 There exists an optimal solution of P3 in which all the delivery shipments, except possibly one, at each plant, are full. More precisely, if there are h_i orders scheduled at plant $i \in M$, where $h_i = ub + v$, for some integers $u \geq 0$ and $0 \leq v < b$, then ub orders are delivered in u full shipments and v orders are delivered in a partial shipment.

Proof Clearly, in problem P3, $D_{\max} = \max\{P_i + t_i | i \in M\}$, where P_i denotes the total processing time of the orders assigned to plant i. Given an assignment of orders to each plant, $D_{\max}$ is independent of how the delivery shipments are formed. Thus given an assignment of the orders to the plants, the problem reduces to minimizing total production and distribution cost TC, which is equivalent to minimizing the total distribution cost because the total production cost is fixed once an order assignment is given. It can be seen that the order delivery schedule given in the statement of the lemma minimizes the total distribution cost for a given order assignment. Thus there exists an optimal overall schedule that follows such a delivery schedule. ■

In the remainder of this section, we propose a linear programming based heuristic for problem P3, analyze the worst-case and asymptotic performance of the heuristic, and evaluate its performance computationally. We first consider a slightly different problem denoted as $P3'$. Everything else in $P3'$ is the same as in P3 except that each order is required to be delivered in a separate shipment and the transportation cost of a shipment from plant $i \in M$ is defined to be f_i/b. In an optimal solution of $P3'$, there may exist some plants where no orders are scheduled due to, for example, very large transportation times t_i and production costs c_{ij} associated with these plants. If we require that all the orders be scheduled on a subset of plants $Q \subseteq M$ only, then $P3'$ can be formulated as the following integer program:

$$IP(Q): \qquad Z(Q) = \min \quad \alpha D_{\max} + (1-\alpha) \sum_{i \in Q} \sum_{j=1}^{n} \left(c_{ij} + \frac{f_i}{b} \right) x_{ij} \tag{2.11}$$

Subject to:

$$D_{\max} \geq \sum_{j=1}^{n} p_{ij} x_{ij} + t_i, \quad \text{for} \quad i \in Q \tag{2.12}$$

$$\sum_{i \in Q} x_{ij} = 1, \quad \text{for} \quad j \in N \tag{2.13}$$

$$x_{ij} \in \{0, 1\} \qquad \text{for} \quad i \in Q, \ j \in N \tag{2.14}$$

where each binary variable x_{ij} is defined to be 1 if order j is assigned to plant $i \in Q$, and 0 otherwise, and variable $D_{\max}$ is the maximum delivery time of the orders. Constraint (2.12) defines $D_{\max}$, and Constraint (2.13) ensures that each order is assigned to one of the plants in Q. It should be noted that Constraint (2.12) implies that there is at least one order scheduled at each plant in Q because otherwise $D_{\max}$ does not have to be greater than or equal to t_i. This means that problem $P3'$ is not equivalent to IP(M). Instead, problem $P3'$ is equivalent to the problem of finding a subset $Q \subseteq M$ with a minimum possible objective value $Z(Q)$.

We are interested in the LP relaxation of $IP(Q)$, denoted as $LP(Q)$. Denote the optimal objective value of $LP(Q)$ by $Z_{LP}(Q)$. Clearly, if we can find a subset $Q \subseteq M$ such that $Z_{LP}(Q)$ is minimum possible, then $Z_{LP}(Q)$ is a lower bound of the optimal objective value of problem $P3'$. We propose the following procedure to find such a subset Q without solving the LP relaxation for every subset of M.

Algorithm LB

Step 0: Reindex the plants in the nonincreasing order of transportation times t_i. Let Q_i be the subset of the plants $\{i, i+1, ..., m\}$.

Step 1: For $i = 1, \ldots, m$, solve $LP(Q_i)$ and get the optimal objective value $Z_{LP}(Q_i)$.

Step 2: Let U denote the subset of plants Q_i with the lowest objective value $Z_{LP}(Q_i)$, and let $u = |U|$.

Algorithm LB solves m linear programs, and hence is polynomial in both n and m.

Theorem 8 The value $Z_{LP}(U)$ generated by Algorithm LB is a valid lower bound of the optimal objective value of problem $P3'$.

Proof We prove this by contradiction. Suppose that a subset of plants $A \subseteq M$, where $A \neq U$, gives a lower objective value $Z_{LP}(A)$ than $Z_{LP}(U)$. Construct a new set of plants A' as follows: Set $t = \max\{t_i | i \in A\}$, and define $A' = A \cup \{i \in M | t_i \leq t\}$. Since $A' = Q_i$ for some i, $LP(A')$ is one of the LPs solved in Algorithm LB. Thus

$$Z_{LP}(U) \leq Z_{LP}(A') \tag{2.15}$$

Since every feasible solution to problem $LP(A)$ is also feasible for problem $LP(A')$, with the same objective value, we have

$$]Z_{LP}(A') \leq Z_{LP}(A) \tag{2.16}$$

From (2.15) and (2.16), we get: $Z_{LP}(U) \leq Z_{LP}(A)$, which contradicts our earlier claim. Therefore, $Z_{LP}(U)$ gives a valid lower bound for $P3'$. ■

Corollary 9 The value $Z_{LP}(U)$ generated by Algorithm LB is a valid lower bound of the optimal objective value of problem P3.

Proof In problem P3, if order $j \in N$ is assigned to plant $i \in M$, then the contribution of order j to the total transportation and production cost is at least $c_{ij} + f_i/b$, whereas in problem $P3'$ that contribution is exactly $c_{ij} + f_i/b$. This means that the optimal objective value of $P3'$ is a lower bound of that of P3. Then, the corollary follows immediately from Theorem 8. ■

Lemma 7 In an optimal basic solution of $LP(U)$, if $n \geq u - 1$, then at least $(n - u + 1)$ x_{ij} variables take the value 1, where U is the subset of the plants found by Algorithm LB and u is the number of plants in U.

Proof Since there are $n + u$ constraints in $LP(U)$ in addition to the non-negativity constraints, in an optimal basic solution (which can be obtained by the simplex method), there are no more than $n + u$ variables which may take positive values. Since variable $D_{\max}$ takes a positive value, there are at most $(n + u - 1)$ x_{ij} variables with a positive value in an optimal basic solution of $LP(U)$. Given an optimal basic solution of $LP(U)$, for $j \in N$, let K_j be the subset of variables in the set $\{x_{ij}, i \in U\}$ with a positive value. Define $N_1 = \{j \in N || K_j| = 1\}$, and $N_2 = \{j \in N || K_j| \geq 2\}$. Constraint (2.13) implies that for $j \in N_1$, the only variable in K_j takes the value 1. Clearly, each K_j contains a distinct set of variables. Thus

$$n + u - 1 \geq |N_1| + 2|N_2| = |N_1| + 2(n - |N_1|) = 2n - |N_1|$$

which implies that $|N_1| \geq n - u + 1$. This establishes the lemma. ■

Now we are ready to describe our heuristic for problem P3. The heuristic generalizes the approach proposed by Potts (1985) for the classical parallel machine maximum completion time scheduling problem, which can be viewed as a special case of our problem with no production costs and distribution times and costs, i.e. $c_{ij} = 0, t_i = 0, f_i = 0$ for all $i \in M$ and $j \in N$. Potts formulates his problem as an integer program which can be viewed as a special case of $IP(M)$, and assigns jobs to machines based on the optimal solution of the LP relaxation of this integer program. Although we follow Potts' idea, our heuristic is not a trivial generalization because our problem P3 is much more complex. Furthermore, since our problem is more general, our analysis of the heuristic performance are different from his.

Heuristic H3

Step 1: Run Algorithm LB to obtain a subset of plants U and an optimal basic solution of $LP(U)$. Define subset of orders $J = \{j \in N | x_{ij} = 1$ for some $i \in U$ in this solution$\}$. Lemma 7 implies that there are at least $n - u + 1$ orders in J and at most $u - 1$ orders in $N \backslash J$.

Step 2: (Create a schedule for orders in J) Assign each order $j \in J$ to plant $i \in U$ with $x_{ij} = 1$. Schedule the orders assigned to each plant in an arbitrary sequence. Schedule order delivery such that it satisfies Lemma 6. Denote the resulting partial schedule (containing the orders from J only) by σ_1.

Step 3:(Create a separate schedule for orders in $N \backslash J$) Enumerate all possible assignments of the no more than $u - 1$ orders in $N \backslash J$ to the plants in U till the following termination condition is satisfied. For each such assignment, schedule the orders at each plant in an arbitrary sequence. Schedule order delivery such that it satisfies Lemma 6. If the total contribution of the orders to the objective value of P3 is less than or equal to $Z_{LP}(U)$, then stop. Denote the resulting partial schedule (containing the orders from $N \backslash J$ only) by σ_2. If no schedule satisfies the termination condition, then take the schedule with the lowest total contribution to the objective value of P3, and denote this partial schedule by σ_2.

Step 4: Concatenate schedules σ_1 and σ_2 at each plant. Reschedule order delivery in the concatenated schedule at each plant such that it satisfies Lemma 6. Denote the final schedule by σ.

We note that Step 3 is done independent of Step 2, and there may exist two partial delivery shipments at each plant $i \in U$ if we just concatenate σ_1 and σ_2 without reoptimizing its delivery schedule in Step 4. The enumeration procedure in Step 3 may generate a maximum of u^{u-1} schedules. Thus the worst-case time complexity of H3 is polynomial in n but exponential in m. However, if m is fixed, H3 is a polynomial-time algorithm. We also note that the termination condition in Step 3 may not always be satisfied. However, for a problem with a large number of orders, $Z_{LP}(U)$ is sufficiently large such that the termination condition may be satisfied at an early stage, and hence only a small number of schedules may be generated in Step 3.

Next we analyze the worst-case and asymptotic performance of heuristic H3. We denote the total contribution of the orders in a schedule π to the objective value of P3 by $F_{P3}(\pi)$, and that to the objective value of $P3'$ by $F_{P3'}(\pi)$. Let F^*_{P3} denote the optimal objective value of P3.

Theorem 10 $F_{P3}(\sigma) \leq (b+1)F^*_{P3}$, i.e. the worst-case performance ratio of Heuristic H3 for problem P3 is no more than $b+1$.

Proof Since σ_1 generated in Step 2 of the heuristic only includes the orders $j \in N$ with $x_{ij} = 1$ for some $i \in M$ in the solution of LP(U), we have $F_{P3'}(\sigma_1) \leq Z_{LP}(U)$. By Corollary 9, $Z_{LP}(U) \leq F^*_{P3}$. Therefore,

$$F_{P3'}(\sigma_1) \leq Z_{LP}(U) \leq F^*_{P3} \tag{2.17}$$

There are two cases associated with σ_2 generated in Step 3 of the heuristic:

Case (i): If σ_2 satisfies the termination condition, then $F_{P3}(\sigma_2) \leq Z_{LP}(U)$. By (2.17), $F_{P3}(\sigma_2) \leq F^*_{P3}$.

Case (ii): If σ_2 does not satisfy that condition, then σ_2 is an optimal schedule for the orders in $N \backslash J$. Since only a subset of orders is involved in σ_2, $F_{P3}(\sigma_2) \leq F^*_{P3}$.

Therefore, in both cases, we have

$$F_{P3}(\sigma_2) \leq F^*_{P3} \tag{2.18}$$

In σ_1, since the delivery schedule satisfies Lemma 6, there is at most one partial shipment at each plant $i \in U$. Define V to be the subset of plants in U where there is a partial delivery shipment. Consider the total contribution of the orders in σ_1 to the objective value of $P3'$ (in this case, each order is delivered as a separate shipment with the delivery cost of each shipment from plant i being f_i/b). We have

$$F_{P3'}(\sigma_1) \geq \sum_{i \in V} f_i/b \tag{2.19}$$

and

$$F_{P3}(\sigma_1) \leq F_{P3'}(\sigma_1) + \sum_{i \in V} (b-1) f_i/b \tag{2.20}$$

Inequality (2.20) holds because in problem P3, the total transportation cost of a partial shipment at plant i is f_i, while under problem $P3'$, the total transportation cost of a partial shipment at plant i is at least f_i/b. By (2.17), (2.19) and (2.20), we have

$$F_{P3}(\sigma_1) \leq F_{P3'}(\sigma_1) + (b-1)F_{P3'}(\sigma_1) = bF_{P3'}(\sigma_1) \leq bZ_{LP}(U) \tag{2.21}$$

Since the concatenated schedule is reoptimized with respect to delivery schedule in Step 4 of the heuristic, $F_{P3}(\sigma) \leq F_{P3}(\sigma_1) + F_{P3}(\sigma_2)$. By (2.17), (2.18) and (2.21), we have

$$F_{P3}(\sigma) \leq bZ_{LP}(U) + F^*_{P3} \leq (b+1)F^*_{P3}.$$

This establishes the theorem. ■

Theorem 11 Solution σ generated by heuristic H3 is asymptotically optimal for problem P3 when n goes to infinity with m and b fixed.

Proof Since $u \leq m$ which is fixed, the contribution of the $u-1$ orders in $N \backslash J$ to the objective value of P3 under any schedule is always bounded from above. On the other hand, the objective value of P3 under any schedule becomes infinity when n goes to infinity. Therefore,

$$\lim_{n \to \infty} \frac{F_{P3}(\sigma_2)}{F^*_{P3}} = 0 \tag{2.22}$$

As discussed in the proof of Theorem 10, there is at most one partial shipment at each plant $i \in U$. Define V to be the subset of plants in U where there is a partial delivery shipment. Since $|V| \leq u \leq m$ and f_i for $i \in V$ are all finite, we have

$$\lim_{n \to \infty} \frac{\sum_{i \in V} f_i}{F^*_{P3}} = 0 \tag{2.23}$$

Thus,

$$\begin{aligned}
\lim_{n \to \infty} \frac{F_{P3}(\sigma) - F^*_{P3}}{F^*_{P3}} &\leq \lim_{n \to \infty} \frac{F_{P3}(\sigma_1) + F_{P3}(\sigma_2) - F^*_{P3}}{F^*_{P3}} \\
&\leq \lim_{n \to \infty} \frac{F_{P3'}(\sigma_1) + \sum_{i \in V}(b-1)f_i/b + F_{P3}(\sigma_2) - F^*_{P3}}{F^*_{P3}} \quad \text{by (2.20)} \\
&\leq \lim_{n \to \infty} \frac{\sum_{i \in V}(b-1)f_i/b + F_{P3}(\sigma_2)}{F^*_{P3}} \quad \text{by (2.17)} \\
&= 0 \quad \text{by (2.22) and (2.23)}
\end{aligned}$$

This establishes the theorem. ■

To evaluate the performance of heuristic H3, we conduct a computational experiment as follows. All the parameters except the weighting parameter in the objective function α

are generated exactly the same way as in the test for H1-BASE and H1-IMP in Section 2.3. Three values of α are tested: $\alpha \in \{0.5, 0.9, 0.99\}$. We found that instances with $\alpha < 0.5$ gave results similar to those with $\alpha = 0.5$, and these three values of α gave a relatively large variety of results. It should be noted that in practice, α is determined by the preference (or utility function) of the decision maker which may vary widely from firm to firm.

Similar to the computational experiment in Section 2.3, for each of the 243 combinations of the five parameters with multiple choices $(n, m, b, \rho_i, \alpha)$, we test 20 randomly generated instances, 10 with c_{ij} generated following the Type-1 scheme, and 10 with c_{ij} generated following the Type-2 scheme. Table 2.3 reports the computational results, where each column represents the same performance measure as the corresponding one in Table 2.1. The relative gap of a test problem is the relative gap between the objective value of the solution generated by heuristic H3 and the lower bound $Z_{LP}(U)$ generated in Step 1 of the heuristic. Since every test problem is solved within a small amount of CPU time, we do not report the CPU times in the table.

From Table 2.3 we can derive that (i) the average gap over all the problems with 50, 100, and 200 orders are $5.23\%, 3.11\%$, and 1.96%, respectively; and (ii) the percent of test problems with 50, 100, and 200 orders that have a relative gap over 10% are $12.8\%, 3.8\%$, and 0.9%, respectively. These results show that the H3 is capable of generating near-optimal solutions for most of the problems tested. Furthermore, its performance improves with the number of orders n in a test problem with fixed m and b. Figures 2.2 and 2.3 show the asymptotic optimality property of H3 with respect to the number of orders.

2.6 Problem P4: Minimizing TC subject to $D_{\max} \le D$

The problem even with two plants (i.e. $m = 2$) is at least ordinarily NP-hard because even finding a feasible solution to the problem with $m = 2$ and $t_i = 0$ for $i \in M$ is as hard as the ordinarily NP-complete Partition problem (Garey and Johnson 1979). We use the idea developed in Section 2.5 to design a heuristic for this problem with a general number of plants. Because of the constraint $D_{\max} \le D$, no orders will be assigned to a plant i with $t_i > D - p_{i,\min}$, where $p_{i,\min} = \min\{p_{ij} | j \in N\}$, and hence such a plant can be removed from M. Therefore, without loss of generality, we assume that $t_i \le D - p_{i,\min}$, for all $i \in M$.

We first consider a slightly different problem denoted as $P4'$. Everything else in $P4'$ is the same as in P4 except that each order is required to be delivered in a separate shipment and the transportation cost of a shipment from plant $i \in M$ is defined to be f_i/b. Problem $P4'$ can be formulated as the following integer program:

$$IP': \qquad Z = \min \quad \sum_{i \in M} \sum_{j=1}^{n} \left(c_{ij} + \frac{f_i}{b} \right) x_{ij} \tag{2.24}$$

Subject to:

$$\sum_{j=1}^{n} p_{ij} x_{ij} + t_i \quad \le \quad D, \quad \text{for} \quad i \in M \tag{2.25}$$

$$\sum_{i \in M} x_{ij} \quad = \quad 1, \quad \text{for} \quad j \in N \tag{2.26}$$

$$x_{ij} \in \{0, 1\} \qquad \text{for} \quad i \in M, \; j \in N \tag{2.27}$$

where binary variable x_{ij} is defined to be 1 if order $j \in N$ is assigned to plant $i \in M$, and 0 otherwise. This formulation is similar to $IP(Q)$ given in Section 2.5 for problem $P3'$ with a given subset of plants Q.

Clearly, if the LP relaxation problem of IP', denoted as LP', is infeasible, then problems $P4'$ and P4 are both infeasible. To avoid this uninteresting case, we assume that LP' is always feasible. Let Z'_{LP} denote the optimal objective value of the LP relaxation problem LP'. Using similar arguments as in the proof of Corollary 9, it can be proved that Z'_{LP} is a valid lower bound for the optimal objective value of problem P4. Also, by similar arguments as in the proof of Lemma 7, it can be shown that in an optimal basic solution of LP', if $n \geq m$, there are at least $(n-m)$ x_{ij} variables taking the value 1, and at most m orders with fractional x_{ij} values. Next we describe the heuristic for problem P4.

Heuristic H4

Step 1: Solve the LP relaxation problem LP' and get an optimal basic solution. Define subset of orders $J = \{j \in N | x_{ij} = 1 \text{ for some } i \in M \text{ in this solution}\}$.

Step 2: Assign each order $j \in J$ to plant $i \in M$ with $x_{ij} = 1$. Schedule the orders assigned to each plant in an arbitrary sequence. Schedule order delivery such that it satisfies Lemma 6. Denote the resulting partial schedule (containing the orders from J only) by σ_1.

Step 3: Enumerate all possible assignments of the no more than m orders in $N \backslash J$ to the plants in M. For each such assignment, let P_i be the total processing times of all the orders assigned to plant $i \in M$ (including the orders in σ_1 which have been scheduled in Step 2). If $P_i + t_i > D$ for some $i \in M$, then this assignment is infeasible and discarded. Otherwise, schedule delivery shipments of all the orders assigned to each plant $i \in M$ (including orders in σ_1) such that the delivery schedule satisfies Lemma 6. If no assignment of the orders in $N \backslash J$ to the plants in M leads to a feasible schedule, then go to Step 4. Otherwise, stop; the feasible schedule generated corresponding to one of the assignments with the lowest total cost is adopted. Let this solution be denoted as σ.

Step 4: Decrease the current D (the right-hand side of (2.25)) by $0.1p_{\max}$, where $p_{\max} = \max\{p_{ij} | i \in M, j \in N\}$. Re-run Step 1 with the updated D. If LP' with the new D value is not feasible, then stop; this heuristic fails to produce a feasible solution. Otherwise, re-run Steps 2 and 3 with the originally given D.

We note that Steps 1 and 2 of H4 are very similar to the same steps of H3. However, unlike Step 3 of H3 where a separate schedule is created for the orders in J, Step 3 of H4 adds orders in J to the schedule σ_1 created earlier. In case no feasible solution is generated in Step 3, Step 4 is used in H4 to generate a new solution of x_{ij} by solving a slightly modified LP relaxation problem. Then Steps 2 and 3 are repeated given this new solution of x_{ij}. The quantity $0.1p_{\max}$ used in Step 4 (by which D is reduced) is heuristically set.

It can be seen that the algorithm has a polynomial time complexity if the number of plants m is fixed. As we noted earlier, even finding a feasible solution for problem P4 is NP-hard. Therefore, any polynomial-time heuristic including H4 may fail to find a feasible solution for P4 even if there is a feasible solution. However, if the given constant D satisfies a certain condition, then H4 always generates a feasible solution to P4. This is stated in the following theorem.

Theorem 12 If the given constant D is such that the LP relaxation problem LP' with the right-hand side of (2.25) replaced by $D - p_{\max}$ is feasible, then it is guaranteed that H4 generates a feasible solution to problem P4.

Proof Suppose that the heuristic has not generated a feasible solution for problem P4 after D has been reduced for nine times, i.e. the current D is the original D minus $0.9p_{\max}$. Since LP' with the right-hand side of (2.25) replaced by $D - p_{\max}$ is feasible, every LP' involved in each of these nine iterations is also feasible. When D is reduced for the tenth time, i.e. D becomes the original D minus $p_{\max}$, since the corresponding LP' is feasible, Step 1 generates an optimal basic solution, denoted as x^0. Let $J_i = \{j \in N | x^0_{ij} = 1\}$, for

$i \in M$. Constraint (2.25) in LP' implies that

$$\sum_{j \in J_i} p_{ij} x_{ij}^0 + t_i \leq D - p_{\max}, \text{ for } \ i \in M$$

This means that after Step 2 is executed based on the solution x^0, the total processing time of the orders assigned to each plant $i \in U$ under schedule σ_1 is no more than $D - p_{\max} - t_i$. Furthermore, as we noted earlier, there are at most m orders in $N \backslash J$ under solution x^0. In Step 3, if we assign each order $j \in N \backslash J$ to a different plant in M, then the total processing time of all the orders (including the orders in σ_1) assigned to a plant $i \in M$ is no more than $D - t_i$. This implies that the solution generated following such an assignment of orders in $N \backslash J$ to the plants is feasible to problem P4. Since Step 3 tries every possible assignment of the orders in $N \backslash J$ to the plants, such a feasible solution to P4 is generated in Step 3. Therefore, the heuristic terminates with a feasible solution to P4. ■

Next we computationally evaluate the performance of H4. All the parameters except D are generated exactly the same way as in the test for H1-BASE and H1-IMP in Section 2.3. Parameter D is given by $D = D^1 + \beta(D^0 - D^1)$ where $\beta \in \{0.25, 0.5, 0.75\}$, and D^1 and D^0 are minimum and maximum possible values of maximum delivery time $D_{\max}$ of orders in a schedule, respectively. The values of D^1 is equal to the total delivery time of the orders in the solution of problem P4 with $\alpha = 1$ obtained by heuristic H3. Similarly, the value of D^0 is equal to the total delivery time of the orders in the solution of problem P4 with $\alpha = 0$ obtained by heuristic H3.

Similar to the computational experiments conducted in earlier sections, for each of the 243 combinations of the five parameters with multiple choices (n, m, b, ρ_i, β), we test 20 randomly generated instances, 10 with c_{ij} generated following the Type-1 scheme, and 10 with c_{ij} generated following the Type-2 scheme. The heuristic successfully generates a feasible solution for each test problem. The results are reported in Table 2.4, where each column represents the same performance measure as the corresponding one in Table 2.1. The relative gap of a test problem is the relative gap between the objective value of the solution generated by heuristic H4 and the lower bound $Z_{LP'}$ (i.e. the optimal objective value of LP') generated in Step 1 of the heuristic. Since every test problem is solved within a small amount of CPU time, we do not report the CPU times in the table.

From the results in Table 2.4, we can see that (i) the average gap over all the problems with 50, 100, and 200 orders are $10.92\%, 5.35\%$, and 3.09%, respectively; and (ii) the percent of test problems with 50, 100, and 200 orders that have a relative gap over 10% are $39.4\%, 15.5\%$, and 4.1%, respectively. These results show that the performance of H4 improves with the number of orders n and that it is capable of generating near-optimal solutions for most test problems with 100 orders or more.

2.6.1 A Note on Problem P4

We note that problem P4 can be formulated as a fairly simple mixed integer program (MIP) as follows, where binary variable x_{ij} is 1 if order $j \in N$ is assigned to plant $i \in M$ and y_i is a positive integer representing the number of shipments from plant $i \in M$.

$$\min \quad \sum_{i \in M} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i \in M} f_i y_i$$

Subject to:

$$\begin{aligned}
\sum_{j=1}^{n} p_{ij} x_{ij} + t_i \quad &\leq \quad D, \quad \text{for} \quad i \in M \\
\sum_{i \in M} x_{ij} \quad &= \quad 1 \quad \text{for} \quad j \in N \\
\sum_{j \in N} x_{ij} \quad &\leq \quad b y_i \quad \text{for} \quad i \in M \\
x_{ij} \in \{0,1\} \qquad & y_i \geq 0, \quad \text{integer} \quad \text{for} \quad i \in M, j \in N
\end{aligned}$$

We tried to solve this formulation directly by calling the MIP solver of CPLEX for various problem sizes. Our test results showed that the CPU time CPLEX requires to solve a problem increases exponentially with the number of orders n, number of plants m, and the shipment size b. CPLEX is capable of finding optimal solutions for problems with up to 100 orders within a few CPU seconds. However, to find an optimal solution for problems with 200 or more orders, CPLEX usually takes a very long time (1000 CPU seconds if $m \leq 8$ and $b \leq 3$, and more than one day if $m \geq 8$ and $b \geq 4$). It can be concluded that solving P4 directly as a mixed integer program by CPLEX is generally not going to work for problems with 200 or more orders. The fast heuristic H4 proposed here can be used to get a near optimal solution for such problems.

2.7 Conclusions

In this chapter, we have studied four problems related to order assignment and scheduling in a supply chain. Computational complexity of various cases of the problems have been clarified, and polynomial-time exact algorithms have been proposed for some special cases of the problems. All of the four problems are in general NP-hard, and fast heuristics have been proposed for each of them. Worst-case and asymptotic performance of two of the heuristics have been analyzed. Each heuristic has been evaluated computationally and the results show that each heuristic is capable of generating near optimal solutions for most test problems with 100 or more orders.

The analyses and solution approaches developed in this chapter can be generalized to certain extensions of our problems. We have assumed that each plant is capable of processing all the orders. In a more general setting, each plant $i \in M$ may only be qualified to produce a subset of orders $N_i \subseteq N$. In this case, all the heuristics developed in the chapter still work after minor modifications. In H1-BASE, we can define $a_{(k,i)j}$ to be a sufficiently large number, instead of the quantity given in (1), for every combination of (k, i, j) with $j \notin N_i$, such that order j will not be assigned to plant i in the solution of the assignment problem in Phase 1 of the heuristic. The results on the worst-case and asymptotic performance of the heuristic are still valid. Similarly, in the formulations $IP(Q)$ and IP' involved in heuristics H3 and H4, respectively, we can define the production cost c_{ij} of order $j \notin N_i$ at plant $i \in M$ to be sufficiently large such that order j will not be assigned to plant i in the solution of the LP relaxation of these formulations. All the results about these heuristics still hold.

We have assumed that the shipment capacity from different plants is identical. In a more general setting, different plants may be associated with different transportation modes, for example, trucks are used for delivering orders from some plants to the DC whereas air freight is used by some others. Hence the capacity of the shipments from different plants may be different. The heuristics proposed in the chapter can be extended fairly easily to this more general case of the problems.

Figure 2.2: Asymptotic optimality behavior for the average gap for H1-IMP and H3

Figure 2.3: Asymptotic optimality behavior for the maximum gap for H1-IMP and H3

Table 2.1: Computational Results of Heuristics H1-BASE and H1-IMP

			H1-BASE			H1-IMP		
n	m	b	Avg Gap	Max Gap		Avg Gap	Max Gap	
50	2	3	0.62%	4.73%		0.51%	4.73%	
		6	0.75%	7.00%		0.65%	6.04%	
		12	0.68%	5.29%		0.62%	3.51%	
50	4	3	1.66%	8.31%		1.40%	6.52%	
		6	2.26%	14.84%	(5)	1.77%	11.32%	(1)
		12	2.27%	13.35%	(6)	2.00%	12.50%	(3)
50	8	3	2.78%	13.85%	(5)	2.09%	10.05%	(1)
		6	4.89%	25.72%	(29)	3.21%	17.44%	(9)
		12	6.60%	36.16%	(44)	3.99%	20.89%	(21)
100	2	3	0.24%	1.52%		0.23%	1.18%	
		6	0.28%	2.17%		0.26%	2.12%	
		12	0.25%	1.77%		0.24%	1.77%	
100	4	3	0.68%	3.83%		0.56%	2.91%	
		6	0.89%	4.77%		0.76%	3.60%	
		12	0.85%	6.56%		0.80%	5.39%	
100	8	3	1.54%	6.92%		1.25%	5.36%	
		6	2.29%	13.61%	(2)	1.77%	9.14%	
		12	2.68%	14.25%	(9)	2.01%	9.74%	
200	2	3	0.10%	0.81%		0.10%	0.81%	
		6	0.11%	0.96%		0.11%	0.96%	
		12	0.10%	0.62%		0.10%	0.61%	
200	4	3	0.36%	2.10%		0.33%	1.64%	
		6	0.42%	2.97%		0.39%	2.97%	
		12	0.38%	2.38%		0.36%	1.97%	
200	8	3	0.80%	3.32%		0.74%	2.96%	
		6	1.11%	6.32%		1.03%	5.00%	
		12	1.19%	7.43%		1.05%	5.18%	

Table 2.2: Computational Results of Heuristic H2

n	m	b	Avg Gap	Max Gap		CPU (Seconds)
50	2	3	3.36%	11.91%	(2)	1
		6	4.09%	19.35%	(10)	1
		12	4.79%	20.14%	(9)	1
50	4	3	5.31%	13.95%	(10)	2
		6	8.44%	26.40%	(53)	2
		12	12.03%	40.26%	(93)	2
50	8	3	7.61%	20.75%	(47)	2
		6	12.65%	36.92%	(109)	2
		12	18.58%	56.25%	(149)	2
100	2	3	1.47%	4.87%		7
		6	1.96%	7.38%		6
		12	2.12%	7.91%		6
100	4	3	2.97%	9.72%		12
		6	4.59%	14.36%	(6)	10
		12	5.37%	23.66%	(20)	10
100	8	3	4.77%	9.90%		14
		6	9.06%	26.86%	(68)	14
		12	12.07%	30.69%	(101)	15
200	2	3	0.81%	2.39%		76
		6	1.22%	3.50%		69
		12	1.43%	6.33%		61
200	4	3	1.77%	4.30%		125
		6	2.72%	7.30%		107
		12	3.64%	10.37%	(2)	115
200	8	3	3.03%	6.61%		131
		6	5.47%	14.21%	(15)	127
		12	7.66%	18.59%	(43)	116

Table 2.3: Computational Results of Heuristic H3

n	m	b	Avg Gap	Max Gap	
		3	1.53%	5.96%	
50	2	6	2.44%	10.93%	(1)
		12	2.99%	11.90%	(5)
		3	3.14%	9.50%	
50	4	6	4.73%	13.41%	(12)
		12	6.27%	18.56%	(36)
		3	4.98%	14.81%	(6)
50	8	6	8.58%	26.08%	(61)
		12	12.38%	45.18%	(87)
		3	0.81%	2.25%	
100	2	6	1.01%	6.27%	
		12	1.26%	7.40%	
		3	1.93%	4.22%	
100	4	6	2.89%	8.32%	
		12	3.69%	13.08%	(3)
		3	3.27%	7.34%	
100	8	6	5.54%	18.58%	(10)
		12	7.60%	26.21%	(49)
		3	0.48%	1.62%	
200	2	6	0.72%	3.17%	
		12	0.86%	4.78%	
		3	1.09%	2.43%	
200	4	6	1.69%	5.35%	
		12	2.18%	7.32%	
		3	1.96%	4.16%	
200	8	6	3.47%	7.45%	
		12	5.20%	14.66%	(15)

Table 2.4: Computational Results of Heuristic H4

n	m	b	Avg Gap	Max Gap	
		3	3.26%	13.78%	(5)
50	2	6	5.30%	13.51%	(21)
		12	7.18%	21.69%	(37)
		3	5.45%	12.84%	(6)
50	4	6	10.51%	23.92%	(92)
		12	14.82%	42.29%	(129)
		3	7.69%	25.29%	(41)
50	8	6	15.93%	45.58%	(136)
		12	28.18%	85.43%	(172)
		3	1.31%	3.59%	
100	2	6	2.03%	6.68%	
		12	2.96%	13.35%	(4)
		3	2.68%	6.83%	
100	4	6	4.91%	11.11%	(6)
		12	6.81%	16.75%	(40)
		3	4.09%	15.35%	(6)
100	8	6	9.08%	25.12%	(69)
		12	14.33%	40.32%	(126)
		3	0.76%	1.95%	
200	2	6	1.24%	4.53%	
		12	1.71%	5.72%	
		3	1.46%	4.91%	
200	4	6	2.74%	6.20%	
		12	4.19%	10.49%	(3)
		3	2.29%	6.70%	
200	8	6	4.95%	12.90%	(4)
		12	8.46%	20.58%	(59)

Chapter 3

Scheduling a Production-Distribution System to Optimize the Tradeoff between Delivery Tardiness and Distribution Cost

3.1 Introduction

We consider a make-to-order production-distribution system consisting of one supplier and one or more customers. At the beginning of a planning horizon, each customer places a set of orders with the supplier. The supplier needs to process these orders and deliver the completed orders to the customers. Each order has a due date specified by the customer. Ideally, each customer wishes to receive her orders from the supplier by their respective due dates. However, since order deliveries incur distribution costs, the supplier wishes to consolidate the order delivery as much as possible to minimize the total distribution cost. Delivery consolidation implies that some completed orders may have to wait for other orders to be completed so that they can be delivered in the same shipment. Hence, some orders may be delivered to their customers after their due dates, resulting in a tradeoff between delivery timeliness and total distribution cost. The problem we consider in this chapter is to find a joint schedule for order processing and delivery so that the tradeoff between the maximum delivery tardiness and total distribution cost is optimized.

This problem is often faced by manufacturers who make time-sensitive products such as fashion apparel and toys, which typically have large product types and sell only during specific seasons. Consider the production and distribution scheduling decisions such a manufacturer needs to make. The customers (e.g. distributors or retailers) often set due dates on the orders they place with the manufacturer and there is typically a penalty imposed on the manufacturer if the orders are not completed and delivered to the customers on time. Hence the manufacturer would like to meet the due dates as much as possible. Another factor the manufacturer has to consider is the total cost for order processing and delivery. Since the products are time-sensitive, orders are delivered shortly after their completion and thus we assume that little inventory cost is incurred. The manufacturer's total cost is mainly contributed by production and distribution operations. The total production cost for a fixed set of orders is normally fixed and independent of the production schedule used. Therefore, the manufacturer should focus on the distribution cost when considering the total cost. Since different orders may have different due dates, delivering more orders on time might require the manufacturer to make a larger number of shipments leading to higher total distribution cost. Therefore, the manufacturer has to find a production and distribution schedule that achieves some balance between delivery timeliness and total distribution cost. In practice, the maximum tardiness of orders and the total tardiness of orders are the two commonly used measurements for delivery timeliness. They represent the worst and average service level with respect to meeting order due dates, respectively. In this chapter, we focus on the maximum tardiness as the measurement for delivery timeliness.

The schematic of the supply chain is given in Fig 3.1. In the following we describe the model to be studied in this chapter. There are one supplier and m ($m \geq 1$) customers $M = \{1, \ldots, m\}$ located at different locations in a given production-distribution system. At the beginning of a planning horizon, the supplier receives n_i orders from each customer

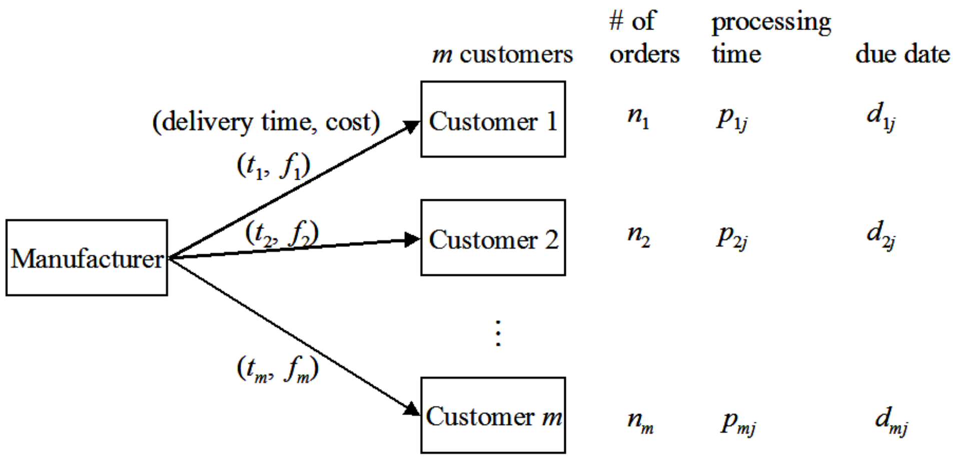

Figure 3.1: The supply chain

$i \in M$, requesting for processing. Let $n = n_1 + \ldots + n_m$ be the total number of orders. Let (i, j) denote the jth order from customer i, $N_i = \{(i, 1), \ldots, (i, n_i)\}$ be the set of the orders from customer i, and $N = N_1 \cup \ldots \cup N_m$ be the set of all the orders. All the orders are to be processed on a single production line at the supplier. Each order $(i, j) \in N$ has a processing time p_{ij} and a due date d_{ij}. Completed orders are delivered in batches to the customers. Due to the time sensitivity of the orders and the fact that each customer is located at a distinct location, direct shipping from the supplier to each customer is used. Therefore, only orders from the same customer can be batched together to form a delivery shipment and orders from different customers must be delivered separately. The delivery time and delivery cost from the supplier to customer $i \in M$ are t_i and f_i, respectively. The maximum allowed batch size (i.e. the maximum number of orders that can be shipped in a batch) is given by b. Let C_{ij} and D_{ij} denote the processing completion time and delivery time of order $(i, j) \in N$, respectively. We define $T_{ij} = \max\{0, D_{ij} - d_{ij}\}$ to be the tardiness of a particular order $(i, j) \in N$ and $T_{\max} = \max\{T_{ij} | (i, j) \in N\}$ to be the maximum tardiness of all orders. The total distribution cost for a given schedule is denoted as G, and $G = f_1x_1 + \ldots + f_mx_m$, where x_i is the number of shipments used to deliver the orders of customer $i \in M$.

The objective function should consider both $T_{\max}$ and G. In order to achieve this, we define a weighting factor $\alpha(0 \leq \alpha \leq 1)$, which is based on the manufacturer's relative preference on the two measurements $T_{\max}$ and G. The objective function is then defined as $\alpha T_{\max} + (1 - \alpha)G$. It can be seen that when α is close to 0, more emphasis is given to the total distribution cost. On the other hand, when α is close to 1, more emphasis is given to $T_{\max}$. In situations where the relative preference on the two measurements $T_{\max}$ and G is difficult to quantify, we can simply solve the problem multiple times with varying values of a and pick one of the resulting solutions with the right level of balance between the two measurements $T_{\max}$ and G.

The remainder of the chapter is organized as follows. In Section 3.2, we analyze the computational complexity of the problem under various cases. We give efficient algorithms for the problem under several special cases and show that the problem under the general case is NP-hard. In Section 3.3, we develop a quick heuristic for solving the general case of the problem. We show that the heuristic is asymptotically optimal with respect to the number of orders. To evaluate the performance of the heuristic, we develop a column generation based approach for generating lower bounds. Our computational experiment shows that the heuristic is capable of generating near optimal solutions. In Section 3.4, we compare the performance of the integrated production-distribution approach with two sequential approaches that treat order processing and order delivery independent of each other. Conclusions and scope for future work are given in Section 3.5.

3.2 Analysis of the Problem Solvability

In this section, we consider various cases of the problem. Since the problem with one customer, i.e. $m = 1$, has a different complexity from the problem with multiple customers, we consider these two cases separately. Another important case of the problem is when processing times and due dates of the orders are agreeable, i.e. if $p_{iu} \leq p_{iv}$, then $d_{iu} \leq d_{iv}$, for $1 \leq u, v \leq n_i$ and $i \in M$. This case arises in many practical environments where order due dates are set as a given multiple of the processing times. We define all the cases of the problem considered in this section as follows:

P1: The case where there is only one customer. In this case, for ease of presentation, we drop the customer index i from the problem parameters. Thus the n orders involved in the problem are $N = \{1, \ldots, n\}$, their processing times and due dates are $p_1, \ldots, p_n$, and $d_1, \ldots, d_n$ respectively, and the transportation time and transportation cost are t and f respectively.

P1A: The case P1 with agreeable processing times and due dates.

P2: The case where there are multiple customers, i.e., $m \geq 2$.

P2A: The case P2 with agreeable processing times and due dates.

Clearly, P1 is more general than P1A, and P2 is more general than the other three cases. We study the solvability of a problem by either providing an efficient algorithm for finding an optimal solution or proving that the problem is intractable. Each of these problems is studied next.

Throughout the remainder of this chapter, we say that a set of orders from the same customer are sequenced in EDD order (i.e. earliest due date first order) if they are sequenced in the non-decreasing order of their due dates. We require that in case of a tie in due dates, the orders are sequenced in the non-decreasing order of their processing times, and that if both the due dates and the processing times are the same among a set of orders, they are sequenced according to their indices. It can be seen that the above tie-breaking rule defines a unique EDD sequence for a given set of orders.

For all the problems, it is assumed without loss of generality that the orders belonging to each customer $i \in M$ have been indexed in the EDD order, i.e. $d_{i1} \leq d_{i2} \leq \ldots d_{in_i}$ for $i \in M$. Also, it is easily seen that there exists an optimal schedule where there is no idle time between the processing of orders at the supplier, and where the orders delivered in the same shipment are processed consecutively at the supplier.

3.2.1 P1A and P2A: The Problems with Agreeable Processing Times and Due Dates

We first present a property of problem P1A.

Lemma 8 There exists an optimal solution for problem P1A where the orders are processed in EDD order.

Proof We prove this lemma by contradiction. Suppose that the lemma does not hold. Then there exist two orders u and v such that u is processed before v and $d_u > d_v$ (and hence $p_u > p_v$). Generate a new schedule by interchanging these two orders, keeping everything else the same as before. If these two orders belong to the same delivery batch in the earlier optimal solution, the value of $T_{\max}$ will remain unchanged, and the new schedule will be equivalent to the old one. Otherwise, the value of $T_{\max}$ will either decrease or remain unchanged because $d_u > d_v$ and $p_u > p_v$. The lemma follows immediately. ■

By a similar argument as in the proof of Lemma 8, we can prove the following result. Hence it is stated without a proof.

Lemma 9 There exists an optimal schedule for problem P2A where the orders from each customer are processed in their respective EDD order.

One of the problems considered by Hall and Potts (2003) can be viewed as a special case of our problem P2 with $b = n$ (i.e. there is no batch size limit), and $t_i = 0$ for all $i \in M$

(i.e. there are no delivery times). They show that the result of Lemma 9 applies to their problem where processing times and due dates are not assumed to be agreeable. Based on this result, they propose an $O(n^{2m+1})$ dynamic programming algorithm for finding an optimal schedule for their problem. The idea of their algorithm is based on the following observation: If the number of delivery batches to each customer in the final schedule is given, then the total transportation cost is fixed. An optimal schedule can then be found by trying out all possible ways of splitting the orders of each customer (sequenced in their EDD order) into a desired number of batches. Their algorithm can be used to solve our problem P2A after it is modified by incorporating the batch size constraint and delivery times into their recursive relations. Since P1A can be viewed as a special case of P2A with $m = 1$, we can conclude that both problems P1A and P2A with a fixed number of customers m can be solved in polynomial time.

As we will see in the next section, when the processing times and due dates are not agreeable, the result of Lemma 8 does not hold for problem P1 and hence the result of Lemma 9 does not hold for problem P2 either. Consequently, the dynamic programming algorithm of Hall and Potts (2003) does not work for our problems P1 or P2.

Next we consider the problem P2A with an arbitrary number of customers. The cases when $\alpha = 0$ or $\alpha = 1$ can be solved very easily. When $\alpha = 0$, the problem can be solved to optimality by minimizing the number of delivery batches for each customer. Any production schedule is optimal. To solve the problem when $\alpha = 1$, we define for each order $(i, j) \in N$, a shipping due date which is the latest time the order (i, j) should leave the supplier in order to reach the customer without any tardiness. An optimal solution to this case is obtained by processing the orders in a non-decreasing sequence of the shipping due dates and delivering each order independently immediately after processing.

We show in the following that when $0 < \alpha < 1$, the problem P2A with an arbitrary m is NP-hard.

Theorem 13 The problem P2A with $0 < \alpha < 1$ and an arbitrary number of customers is NP-hard.

Proof We prove this by reducing the Subset Sum problem, a known NP-hard problem, to P2A. The Subset Sum problem can be stated as follows (Garey and Johnson 1979): Given a set of $v+1$ positive integers $a_1, \ldots, a_v$, and B, does there exist a subset $U \subseteq V = \{1, \ldots, v\}$ such that $\sum_{i \in U} a_i = B$?

Define $A = \sum_{i \in V} a_i$ and $H = \lceil 2v(1-\alpha)/\alpha \rceil + v + B$. We construct a corresponding instance of problem P2A as follows.

Number of customers, $m = v$.

Number of orders from each customer, $n_i = 3$, for $i = 1, \ldots, m$.

Processing times: $p_{i1} = 1, p_{i2} = a_i, p_{i3} = Ha_i$, for $i = 1, \ldots, m$.

Due dates: $d_{i1} = d_1 = m + B, d_{i2} = d_2 = m + B + (H+1)(A-B)$, and

$$d_{i3} = d_3 = m + (H+1)A, \text{ for } i = 1, \ldots, m.$$

Transportation times and costs, $t_i = 0$ and $f_i = 1$ for $i = 1, \ldots, m$.

Maximum allowed batch size, $b = 3$.

Threshold cost, $F = 2m(1-\alpha)$

Clearly, the orders in this instance of P2A have agreeable processing times and due dates. For ease of presentation, we call orders $(i,1), (i,2)$ and $(i,3)$ from each customer $i \in M$ Type I, Type II, and Type III orders, respectively. We first prove the following properties: In a solution to this instance of P2A with the objective value not exceeding F, (a) Type I and Type III orders from each customer will be in different delivery batches, (b)

There will be exactly $2m$ delivery batches, (c) There will be no tardy orders, (d) Orders of Type II will always be in a delivery batch of size 2. The proof for each is given next:

(a)If a Type I order is batched with a Type III order, the tardiness of the Type I order will be more than $H - d_1$, and hence $\alpha T_{\max} > F$, which means that the objective value exceeds F.

(b)If there are more than $2m$ batches, the objective value will be greater than $F = (1-\alpha)2m$. Since Type I and Type III orders cannot be put in the same batch, we need at least $2m$ total batches. Hence we have exactly $2m$ batches.

(c)Since $2m$ batches account for the entire threshold value, there can be no contribution from the tardiness part.

(d)This follows directly from parts (a) and (b).

Now we prove that there is a solution to the constructed instance of P2A with total cost not exceeding F if and only if there is a solution to the instance of the Subset Sum problem.

(If part) Given a subset $U \subseteq V$ such that $\sum_{i \in U} a_i = B$, we construct a schedule for the instance of P2A as follows: First process all the Type I orders from customers $i \in V \backslash U$ and deliver each of them in a separate shipment. Next process the Type I and Type II orders from customers $i \in U$ and deliver them in batches of two orders. Next process the Type II and Type III orders from the customers $i \in V \backslash U$ and deliver them in batches of two orders. Finally process the orders of Type III from the customers $i \in U$ and deliver each of them separately. Let the cardinality of set U be k. The cardinality of $V \backslash U$ is $m-k$. The delivery time of the last batch with Type I and Type II orders is:

$$D_1 = m + \sum_{i \in U} a_i = m + B = d_1$$

Hence all the Type I orders are delivered before or on their due date. Similarly, the delivery time of the last batch with Type II and Type III orders is:

$$D_2 = D_1 + (H+1)\sum_{i \in V \setminus U} a_i = m + B + (H+1)(A-B) = d_2$$

Therefore, all the Type II orders are delivered before or on their due date. Similarly, the delivery time of the last Type III order in the schedule is:

$$D_3 = D_2 + H\sum_{i \in U} a_i = m + B + (H+1)(A-B) + HB = m + (H+1)A = d_3$$

We see that all the orders are delivered on time. The number of batches is given by: $(m-k) + k + (m-k) + k = 2m$. Hence the total cost is F.

(Only if part) Let us assume that there are k batches that consist of orders of Type I and II. Let the indices of the corresponding Type II orders form set $U \subseteq V$. Since no tardy orders exist in the schedule (Result (c) proved earlier), all these orders should be delivered no later than $d_1 = m + B$. Therefore, the maximum delivery time for orders of Type I is: $D_1 = m + \sum_{i \in U} a_i \leq m + B$, which means that

$$\sum_{i \in U} a_i \leq B \tag{3.1}$$

Similarly, the delivery time D_2 of the last batch with Type II and Type III orders should not be greater than $d_2 = m + B + (H+1)(A-B)$. Therefore,

$$D_2 = D_1 + (H+1)\sum_{i \in V \setminus U} a_i = m + \sum_{i \in U} a_i + (H+1)\sum_{i \in V \setminus U} a_i = m + B + (H+1)(A-B)$$

This means that $m + A + H\sum_{i \in V \setminus U} a_i \leq m + HA - HB + A$, which further implies that $H\sum_{i \in V \setminus U} a_i \leq H(A-B)$. Therefore,

$$\sum_{i \in U} a_i \geq B \tag{3.2}$$

From (3.1) and (3.2), we see that $\sum_{i\in U} a_i = B$ must hold. This means that set U is a solution to the instance of the Subset Sum problem.

Combining the "If" part and "Only If" part, we have proved the theorem. ■

Theorem 13 implies that P2, the general case of the problem, is also NP-hard when $0 < \alpha < 1$ and the number of customers is arbitrary. When $\alpha = 0$ or $\alpha = 1$, P2 can be solved in polynomial time by adopting the same approach as the ones described earlier for P2A under those cases. The complexity of P2 with a fixed number of customers and $0 < \alpha < 1$ is an open problem.

3.2.2 P1: The Problem with One Customer and General Processing Times and Due Dates

We note that processing orders in their EDD order at the supplier is not necessarily optimal for this case of the problem. This is illustrated through the following example: Consider 4 orders with the following processing times and due dates: $p_1 = 1, p_2 = 5, p_3 = 1$, and $p_4 = 5$; $d_1 = 2, d_2 = 12, d_3 = 13$, and $d_4 = 14$. The transportation time $t = 0$ while the transportation cost $f = 10$. The maximum allowed batch size $b = 2$. Set $\alpha = 0.5$. Clearly, any solution to this problem will contain at least two delivery batches. Therefore, the objective value cannot be less than 10. We can obtain exactly 10 by putting the first and third orders in the first batch, and the rest in the second. Also, it can be seen that this is the only batch configuration that will give an objective value of 10. But this configuration violates the EDD rule. Hence we conclude that the EDD rule is not necessarily optimal for the general problem P1.

To solve P1, we first consider two related problems, called auxiliary problem one (AP1), and auxiliary problem two (AP2). We will solve our problem P1 by solving AP2 multiple times, where AP2 is solved by solving AP1 multiple times. Suppose that each order $j \in N$

has a deadline e_j. Problem AP1 is to schedule the production and distribution of the orders such that a minimum number of delivery shipments are used and all the orders are delivered to the customer before or at their deadlines. Problem AP2 is to schedule the production and distribution of the orders to minimize their maximum tardiness $T_{\max}$ subject to the constraint that the number of delivery shipments is no more than h for a given integer h. Later when we use AP1 to solve AP2, we will specify the deadline of each order e_j to be d_j plus some allowed tardiness. We focus on AP1 first. We propose the following algorithm to solve this problem. The algorithm schedules orders backwards and forms the delivery shipments from the last to the first.

Algorithm A1

Step 0: Let the set of unscheduled orders be $U = N$. Set the departure time of the current last shipment to be $Q = \sum_{j \in N} p_j$. Let $k = 1$.

Step 1: Find the subset of the orders that can be delivered in the kth last shipment without violating their deadlines, $S = \{j \in U | Q + t \leq e_j\}$. If S is empty but U is not, then stop, and the problem is infeasible.

Step 2: If $|S| > b$, select the b orders with the largest processing times from S. Otherwise, select all the orders from S. Let X and P be the set of the selected orders and the total processing time of these orders, respectively. Process the selected orders consecutively without idle time in the time period $[Q - P, Q]$. Deliver them together in the kth last shipment with departure time Q. Update $Q = Q - P$, and $U = U \backslash X$. If U is empty, then stop, and we have a feasible schedule that uses exactly k batches. Otherwise, let $k = k + 1$, and go to Step 1.

Lemma 10 Algorithm A1 finds an optimal solution to the problem AP1 in $O(n^2 \log n)$ time.

Proof We prove this by showing that any optimal solution π^* to the problem AP1 can be transformed to a feasible solution π generated by this algorithm without increasing the number of shipments. Suppose that for some integer $h \geq 0$, the kth last shipment in π^* is exactly the same as the kth last shipment in π, for $k = 1, \ldots, h$, but the $(h+1)$st last shipment in π^* is different from the $(h+1)$st last shipment in π. This implies that the set of the unscheduled orders U before the $(h+1)$st last shipment is formed is the same in these two solutions. Also, the departure time Q of the $(h+1)$st last shipment is the same in these two solutions. Let $S = \{j \in U | Q + t \leq e_j\}$. There are two cases to consider as follows:

Case (i): $|S| > b$. In this case, the $(h+1)$st shipment in π contains b orders. If there are less than b orders in the $(h+1)$st shipment in π^*, we can move some orders in S that are scheduled in earlier shipments to the $(h+1)$st last shipment so that this shipment contains b orders. If there are b orders in the $(h+1)$st shipment in π^*, we can interchange some orders in this shipment with some other orders in S that are scheduled in earlier shipments but with larger processing times. It can be seen that in both cases the resulting new solution is still feasible and the number of shipments is not increased. Thus the $(h+1)$st shipment in π^* can be transformed such that it becomes exactly the same as the $(h+1)$st shipment in π.

Case (ii): $|S| \leq b$. It can be similarly proved that in this case we can also transform the $(h+1)$st shipment in π^* such that it becomes exactly the same as the $(h+1)$st shipment in π without increasing the number of shipments.

This shows that the solution found by the algorithm is optimal. The algorithm carries out at most n iterations, each consisting of Steps 1 and 2. Since Step 1 takes at most $O(n)$

time and Step 2 takes at most $O(n \log n)$ time, the overall complexity of the algorithm is bounded by $O(n^2 \log n)$. ■

Next we consider the second auxiliary problem AP2 which is to schedule the production and distribution of the orders to minimize their maximum tardiness $T_{\max}$ subject to the constraint that the number of delivery shipments is no more than h for a given integer h, where $\lceil \frac{n}{b} \rceil \leq h \leq n$. It can be seen that the value of $T_{\max}$ is non-increasing with the value of h in the optimal solution of this problem. Based on this observation, we propose the following line search algorithm to find the optimal $T_{\max}$ given h.

Algorithm A2

Step 0: Let T^{LB} and T^{UB} denote a lower bound and an upper bound of the maximum delivery tardiness $T_{\max}$ of orders respectively. Initially, let T^{LB} be the maximum tardiness of orders if they are processed in the EDD order and each order is delivered separately, and let T^{UB} be the maximum tardiness of orders if they are processed in the EDD order and all are delivered in full shipments except possibly the last several orders which may be delivered in a partial shipment. Clearly, $T^{LB} \geq 0$ and $T^{UB} \leq t + P$, where $P = p_1 + \ldots + p_n$.

Step 1: Let $T^0 = \frac{1}{2}(T^{LB} + T^{UB})$. Define auxiliary problem one AP1 by imposing a deadline on each order $j \in N, e_j = d_j + T^0$. Solve this problem by Algorithm A1, and let the optimal number of shipments used be k.

Step 2: If $k > h$, let $T^{LB} = T^0$. Otherwise, let $T^{UB} = T^0$. If $T^{UB} - T^{LB} < 1$, stop. The only integer in the interval $[T^{LB}, T^{UB}]$ is adopted as the solution value of $T_{\max}$. Otherwise, go to Step 1.

Lemma 11 Algorithm A2 finds an optimal solution to the problem AP2 in $O(n^2(\log n)(\log(P+t)))$ time, where $P = p_1 + \ldots + p_n$.

Proof As we observed earlier, the value of $T_{\max}$ is non-increasing with the value of h in the optimal solution of this problem. Thus the solutions T^0 found in the line search involved in the algorithm are guaranteed to converge to the optimal solution if an infinitely many iterations are carried out. However, since the optimal value of $T_{\max}$ must be an integer, there is no need to carry out an infinite number of iterations. As soon as the gap between T^{LB} and T^{UB} becomes less than 1, there is at most one integer that can be contained in the interval $[T^{LB}, T^{UB}]$. Since this line search guarantees that the interval $[T^{LB}, T^{UB}]$ at each iteration contains the optimal solution, there must be an integer in this interval even when the width of this interval is less than 1. This shows that the algorithm does find the optimal solution.

The number of iterations in the line search is bounded by $O(\log(P+t))$. Since it takes $O(n^2 \log n)$ time to run Algorithm A1 in each iteration, the overall computational time is bounded by $O(n^2(\log n)(\log(P+t)))$. ■

We propose the following algorithm based on A2 for solving our problem P1.

Algorithm A3

Step 1: For $h = \lceil \frac{n}{b} \rceil, \ldots, n$, do the following: Define an auxiliary problem AP2 with the number of delivery shipments no more than h. Solve the problem by applying Algorithm A2. Let π_h and $T_{\max}(\pi_h)$ be the optimal schedule and its maximum tardiness found by the algorithm.

Step 2: Find u such that $\alpha T_{\max}(\pi_u) + (1-\alpha)uf = \min\{\alpha T_{\max}(\pi_h) + (1-\alpha)hf | h = \lceil \frac{n}{b} \rceil, \ldots, n\}$. Then schedule π_u is optimal to problem P1 with the objective value $\alpha T_{\max}(\pi_u)+$

$(1-\alpha)uf$.

Theorem 14 Algorithm A3 finds an optimal solution to the problem P1 in $O(n^3(\log n)(\log(P+t)))$ time, where $P = p_1 + \ldots + p_n$.

Proof By the definition of problem AP2, it can be seen that problem P1 is equivalent to finding the best h such that $\alpha T_{\max}(\pi_h) + (1-\alpha)hf$ is minimum. This shows the optimality of the Algorithm A3. Since at most n auxiliary problems AP2 are solved in the algorithm, by Lemma 11, the overall complexity of A3 is bounded by $O(n^3(\log n)(\log(P+t)))$ time. ■

Since the input size of our problem P1 is at least $\sum_{j\in N} (\lceil \log p_j \rceil + \lceil \log d_j \rceil) + \lceil \log t \rceil \geq n + \log(P+t)$, Algorithm A3 is polynomial.

3.3 A Heuristic for the Problem with Multiple Customers when $0 < \alpha < 1$

In this section, we propose and evaluate a heuristic for the problem P2. Since P2 is a more general case of P2A, the heuristic is also applicable to P2A. We first prove an optimality property. Then we develop the heuristic and prove it to be asymptotically optimal. We propose a linear programming based approach for obtaining tight lower bounds and use column generation to solve the linear programming formulations to optimality. A set of computational experiments is carried out to evaluate the performance of the heuristic.

3.3.1 An Optimality Property

We define the SEDD sequence for a given set of orders as follows. Consider the shipping due dates of the orders as defined in Section 3.2. Arrange the orders in the non-decreasing order of their shipping due dates. In case of a tie, arrange the orders in the non-decreasing order of their processing times. If both the shipping due dates and the processing times are the same, arrange them by their customer index followed by their order index. It can be seen that the above tie-breaking rule defines a unique SEDD sequence for a given set of orders. Also, in the SEDD sequence, orders from the same customer are sequenced in their EDD order.

Lemma 12 There exists an optimal schedule for P2 where: (i)The orders that are delivered in the same batch are processed in their EDD sequence at the supplier; (ii) The first orders of the batches form an SEDD sequence; (iii)Let u denote the first order processed in a particular batch. All the orders that come before u in the SEDD sequence of all the orders of N are processed before this batch of orders.

Proof (i) Since all the orders in a batch are delivered at the same time, the tardiness of the batch is not influenced by the processing sequence of the orders in them. So we choose a sequence that is in the EDD order for the set of orders in the batch.

(ii) By (i), we can assume that the orders in each batch are processed in the EDD sequence. Let u be the first order in a batch. Suppose there exists an order v that is the first order in some earlier batch and has a shipping due date larger than that of u. Then we can move the batch containing order v to a position immediately after the batch containing order u without increasing the objective value. We can do this for every pair of batches that violates the lemma. In cases where two batches have their first orders with the same shipping due dates, the relative sequence of these two batches does not affect the tardiness value. Hence there exists an optimal solution where the first orders in the batches reflect the unique SEDD sequence.

(iii) This follows directly from (i) and (ii) ■

3.3.2 The Heuristic

The heuristic first solves a single-customer auxiliary problem for each customer independently in such a way that the contribution due to the other customers is taken care of indirectly. Then it puts together the schedules for individual customers to obtain a combined schedule. We make use of Lemmas 9 and 12. That is, schedules are built such that the orders from each customer follow the EDD sequence and the set of first orders from every batch follows the SEDD sequence. Although Lemma 9 is not valid for the general problem P2, we will show that forcing the EDD sequence for each customer does not affect the results significantly when the number of orders is large.

Define C_{ij}^{SEDD} as the completion time of order $(i,j) \in N$ at the supplier when all of the orders from N are processed in the SEDD sequence. The single-customer auxiliary problem for customer $i \in M$, denoted as AUX_i, is defined as follows: Schedule the processing and delivery of the orders from N_i subject to the following two constraints: (i) the orders are processed in the EDD order at the supplier and (ii) the departure time of each delivery batch B containing (i,j) as the first order is required to be the sum of C_{ij}^{SEDD} and the total processing time of the remaining orders in the batch, i.e. $\sum_{(i,u)\in B} p_{iu} - p_{ij}$. The objective of problem AUX_i is to minimize the maximum delivery tardiness of the orders given that the orders are delivered in a given number of delivery batches. Due to Constraint (ii), a feasible schedule to AUX_i may contain idle time between the processing of the last order in one batch and the first order in the next batch. We present the heuristic next.

Heuristic H1

Step 1: Create an auxiliary problem AUX_i, as described earlier, for each customer $i \in M$. Solve AUX_i, for $i \in M$, by the following dynamic programming algorithm, denoted as DP1, where the value function $F(j,k)$ is defined to be the minimum value of the maximum tardiness for the first j orders $\{(i,1),(i,2),\ldots,(i,j)\}$ when they are delivered in k batches.

DP1

Initial values: $F(0,0) = 0$.

Recursive relations: For $j = 1, \ldots, n_i$, and $k = \lceil \frac{j}{b} \rceil, \ldots, j$,

$$F(j,k) = \min_{1 \le q \le \min\{b,j\}} \left\{ \max \left\{ F(j-q, k-1), \max \left\{ 0, C_{i,j-q+1}^{SEDD} + \sum_{u=j-q+2}^{j} p_{iu} + t_i - d_{i,j-q+1} \right\} \right\} \right\} \quad (3.3)$$

Let $T_{\max}^i(k) = F(n_i, k)$ denote the maximum tardiness for customer i when the orders of customer i are delivered in k batches. Let $\Lambda_i(k)$ denote the corresponding batch configurations for customer i. Let $\Gamma_i = \{T_{\max}^i(k) | k = \lceil \frac{n_i}{b} \rceil, \ldots, n_i\}$, and $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \ldots \Gamma_m$. Clearly, $|\Gamma| \le \sum_{i=1}^m (n_i - \lceil n_i/b \rceil + 1)$.

Step 2: For each $x \in \Gamma$, and each customer $i \in M$, define $k_i(x) = \min\{k \in \{\lceil n_i/b \rceil, \ldots, n_i\} \, | T_{\max}^i(k) \le x\}$ if there exists some $k \in \{\lceil n_i/b \rceil, \ldots, n_i\}$ with $T_{\max}^i(k) \le x$, and $k_i(x) = \infty$ otherwise. Find $x^* \in \Gamma$ such that

$$\alpha x^* + (1-\alpha) \sum_{i \in M} f_i k_i(x^*) = \min_{x \in \Gamma} \left\{ \alpha x + (1-\alpha) \sum_{i \in M} f_i k_i(x) \right\} \quad (3.4)$$

and the corresponding batch configurations $\Lambda_i(k_i(x^*))$ for each customer $i \in M$. Let π_i denote the schedule for customer i corresponding to the value function $F(n_i, k_i(x^*))$ (Note that π_i is optimal to the problem AUX_i with $k_i(x^*)$ delivery batches).

Step 3: Sequence all the batches determined by the batch configurations $\{\Lambda_i(k_i(x^*)) | i \in M\}$ obtained in Step 2 such that the first orders of the batches form the SEDD sequence. This gives a feasible schedule π for the original problem. Calculate the objective value of π.

In the algorithm DP1, the recursive relation (3.3) enumerates all possible sizes q of the current last delivery batch. Hence DP1 solves AUX_i optimally for all possible number of delivery batches k. In Step 2, the selected value of maximum tardiness x^* optimizes the overall objective when each customer is considered separately.

Next we estimate the time complexity of the heuristic. For each customer $i \in M$, the time required by DP1 is $O(n_i^2 b)$. Thus the overall time needed in Step 1 of the heuristic is $O(n^2 b)$. In Step 2, there are no more than n values in the set Γ. For each value $x \in \Gamma$, $k_i(x)$ for customer $i \in M$ can be found by doing a line search for $T_{\max}^i(k)$ corresponding to values of k between $\lceil n_i/b \rceil$ and n_i. This takes $O(\log n_i)$ steps. Therefore, the total complexity of this for all the customers is $O(\sum_{i \in M} \log n_i)$ which is bounded by $O(m \log n)$. Step 3 requires $O(n \log n)$ steps. Hence the overall complexity of the heuristic is bounded by $O(n^2 b + nm \log n)$.

Lemma 13 Denote the optimal objective value of the problem P2 as F*. Then

$$F^* \geq \alpha x^* + (1-\alpha) \sum_{i \in M} f_i k_i(x^*) - \alpha \left((b-1)p_{\max} + 2t_{\max} - 2t_{\min}\right) \tag{3.5}$$

where x^* is as defined in (3.4), $p_{\max} = \max\{p_{ij} | (i,j) \in N\}$, $t_{\max} = \max\{t_i | i \in M\}$, and $t_{\min} = \min\{t_i | i \in M\}$.

Proof Given an optimal schedule S^* of the problem P2 that follows Lemma 12, we construct a schedule S' such that there are same number of delivery batches in S' as in S^*, and each batch in S' contains the same number of orders from the same customer as in the corresponding batch in S^*. But in S' , the orders from each customer are scheduled in their EDD sequence. So the actual set of orders in any batch in S' may be different from that in the corresponding batch in S^*. Each batch in S' is shipped at a time that is the sum of the completion time C_{ij}^{SEDD} of the first order (i,j) in the batch and the total processing time of the other orders in the batch. Note that schedule S' may not be feasible to P2 because there may be overlap between batches of orders from different customers. Also, note that schedule S' gets enumerated implicitly in Step 1 of the heuristic. We can easily see that the total distribution cost in S' is the same as that in S^*.

Let $D_{ij}(S')$ and $D_{ij}(S^*)$ denote the delivery time of order (i,j) in S' and S^* respectively. Similarly, let $T_{ij}(S') = \max\{0, D_{ij}(S') - d_{ij}\}$ and $T_{ij}(S^*) = \max\{0, D_{ij}(S^*) - d_{ij}\}$ denote the tardiness of order (i,j) in S' and S^* respectively. The maximum tardiness of orders in S' is determined solely based on the first order in each batch. Consider the first order, denoted as (i,u) in a particular batch of orders from customer $i \in M$ in schedule S'. Let τ denote the sum of processing times of all the orders except (i,u) in the batch. Evidently, $\tau \leq (b-1)p_{\max}$. Let Q denote the set of all orders up to and including order (i,u) in the SEDD sequence of N. Clearly, $C_{iu}^{SEDD} = \sum_{(i,j)\in Q} p_{ij}$. Thus we have:

$$D_{iu}(S') = C_{iu}^{SEDD} + \tau + t_i = \sum_{(i,j)\in Q} p_{ij} + \tau + t_i \leq \sum_{(i,j)\in Q} p_{ij} + (b-1)p_{\max} + t_{\max} \quad (3.6)$$

Now consider the last batch in S^* that contains an order from Q. Suppose that this batch belongs to customer $k \in M$. Denote the first order in this batch as (k,v). Note that (k,v) belongs to Q. We have the following:

$$D_{kv}(S^*) \geq \sum_{(i,j)\in Q} p_{ij} + t_{\min} \quad (3.7)$$

From (3.6) and (3.7), we have:

$$D_{iu}(S') - D_{kv}(S^*) \leq (b-1)p_{\max} + t_{\max} - t_{\min} \quad (3.8)$$

Since (i,u) is the last order in Q, the shipping due dates follow the relation: $d'_{iu} \geq d'_{kv}$. Therefore, $d_{iu} - t_i \geq d_{kv} - t_k$, which implies that $d_{iu} - d_{kv} \geq t_{\min} - t_{\max}$. This, along with (3.8), implies that

$$(D_{iu}(S') - d_{iu}) - (D_{kv}(S^*) - d_{kv}) \leq (b-1)p_{\max} + 2t_{\max} - 2t_{\min} \quad (3.9)$$

Inequality (3.9) implies that

$$T_{iu}(S') - T_{kv}(S^*) \leq (b-1)p_{\max} + 2t_{\max} - 2t_{\min} \quad (3.10)$$

Inequalities (3.8) through (3.10) are valid for the first order (i,u) of any batch in S'. DP1 in the heuristic H1 considers all choices of batch configurations at every customer including the configurations that appear in schedule S'. In Step 2 of H1, $\alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*)$ is obtained by putting together these values in such a way that the combined value for all the customers is minimum. Therefore, this value will not be greater than the one obtained by combining the values at individual customers in S'. Hence,

$$\alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*) \leq F^* + \alpha((b-1)p_{\max} + 2t_{\max} - 2t_{\min}) \tag{3.11}$$

This completes the proof. ■

Lemma 14 Let F^{H1} represent the objective value of the schedule π obtained by the heuristic H1. Then,

$$F^{H1} \leq \alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*) + \alpha(b-1)(m-1)p_{\max} \tag{3.12}$$

Proof Let us consider an arbitrary batch ω of customer $i \in M$ in the schedule π generated in Step 3 of the heuristic. Denote the first order in the batch as (i,u). Let $D_{iu}(\pi_i)$ and $D_{iu}(\pi)$ denote the delivery times of this batch in the schedule π_i generated in Step 2 and in the schedule π respectively. By the definition of problem AUX_i and the fact that pi_i is optimal to the problem AUX_i with $k_i(x^*)$ delivery batches, we have:

$$D_{iu}(\pi_i) = C_{iu}^{SEDD} + \sum_{(i,j)\in\omega} p_{ij} - p_{iu} + t_i \tag{3.13}$$

Since in schedule π, the first orders of the batches form SEDD sequence and the orders from each customer are sequenced in EDD order, there can be at most one batch scheduled before ω from every customer other than i, that contains orders which come after (i,u) in the SEDD sequence of all the orders of N. Even in those batches, there should be at least one order in each batch that comes before order (i,u) in the SEDD sequence of all the orders of N. Therefore, we have:

$$D_{iu}(\pi) \le \left(C_{iu}^{SEDD} + \sum_{(i,j)\in\omega} p_{ij} - p_{iu} + t_i \right) + (b-1)(m-1)p_{\max} \tag{3.14}$$

Let $T_{iu}(\pi_i)$ and $T_{iu}(\pi)$ denote the tardiness of the batch ω in schedules π_i and π, respectively. Then by (3.13) and (3.14), we have,

$$T_{iu}(\pi) - T_{iu}(\pi_i) \le D_{iu}(\pi) - D_{iu}(\pi_i) \le (b-1)(m-1)p_{\max} \tag{3.15}$$

Here the first relation is not an equality to take into account cases where the batch is delivered before its due date. Relation (3.15) is valid for all the batches and hence $(b-1)(m-1)p_{\max}$ gives an upper bound on the difference in maximum tardiness possible between the schedules π_i and π, for every $i \in M$. Since in the schedule π, the batch configurations $\Lambda_i(k_i(x^*))$ generated in Step 2 is used for each customer $i \in M$, the total distribution cost incurred by the orders of N_i in π is exactly $f_i k_i(x^*)$, which is the same as in schedule π_i. Since the sum of the objective values of the schedules π_i over all $i \in M$ is $\alpha x^* + (1-\alpha)\sum_{i\in M} f_i k_i(x^*)$, the objective value F^{H1} of the schedule π satisfies (3.12). This completes the proof. ■

Theorem 15 If order processing times p_{ij}, delivery times t_i, and delivery costs f_i are drawn from distributions over finite intervals $[L_1, U_1], [L_2, U_2]$, and $[L_3, U_3]$, respectively, with $0 < L_1 \le U_1 < \infty$, $0 < L_2 \le U_2 < \infty$, and $0 < L_3 \le U_3 < \infty$, then the solution generated by the heuristic H1 is asymptotically optimal for problem P2 with $0 < \alpha < 1$ when n goes to infinity, with m and b fixed.

Proof As in the proofs of Lemmas 13 and 14, let F^* and F^{H1} represent the optimal objective value of the problem P2, and the objective value of the schedule π generated by the heuristic respectively. Combining the inequalities (3.5) and (3.12), we have,

$$
\begin{aligned}
F^{H1} &\leq F^* + \alpha((b-1)p_{\max} + 2t_{\max} - 2t_{\min}) + \alpha(b-1)(m-1)p_{\max} \\
&= F^* + \alpha((b-1)mp_{\max} + 2t_{\max} - 2t_{\min})
\end{aligned}
$$

which means that,

$$
\frac{F^{H1} - F^*}{F^*} \leq \frac{\alpha((b-1)mp_{\max} + 2t_{\max} - 2t_{\min})}{F^*} \tag{3.16}
$$

For fixed b, the total distribution cost, and therefore F^*, increases to infinity as the number of orders n increases to infinity. Since $p_{\max} < \infty, t_{\max} < \infty, \alpha < 1$, and m and b are fixed, $\alpha((b-1)mp_{\max} + 2t_{\max} - 2t_{\min})$ is finite. Hence, by (3.16), we have,

$$
\lim_{n\to\infty} \frac{F^{H1} - F^*}{F^*} \leq \lim_{n\to\infty} \frac{\alpha((b-1)mp_{\max} + 2t_{\max} - 2t_{\min})}{F^*} = 0
$$

This shows the theorem. ■

When a batch is being formed for a particular customer, heuristic H1 ignores the effect due to the batches formed for the orders of all other customers. This leads to an increase in the tardiness value when we move from Step 2 to Step 3. One way to limit this increase is to reduce the maximum allowed batch size for a few customers and then run the heuristic again. Doing this may lead to an increase in the number of delivery batches for these customers. But on the other hand, it may also lead to smaller batches being formed, which helps reduce the maximum tardiness value. If we reduce the maximum allowed batch size for those customers that have low transportation costs, the reduction in the maximum tardiness may outweigh the increase in the total distribution cost. Another way to improve an existing solution is to replace the order completion times (C_{ij}^{SEDD}) in Step 1 with the actual order completion times obtained using the heuristic. In doing so, we replace the hypothetical completion time values of Step 1 with something that is more realistic and accounts for the batching of orders. These values can be used to obtain the sequences for Step 2 and subsequently Step 3. This approach favors the formation of smaller batches whenever the delay due to batching starts to accumulate. We include these two procedures as improvement schemes while implementing the heuristic.

In the next section, we describe how to obtain a tight lower bound for the problems P2 and P2A using column generation.

3.3.3 Evaluating the Heuristic

To evaluate the performance of heuristic H1, we need to obtain tight lower bounds. Though Lemma 13 provides a valid lower bound of the optimal objective value of problem P2, our computational tests show that this lower bound is very loose. In this section, we present a linear programming based procedure for obtaining tight lower bounds. We will give an IP formulation for a problem closely related to P2 and describe a procedure for obtaining valid lower bounds using the LP relaxation of this IP formulation.

A sequential search approach for obtaining lower bounds

We first consider a closely related problem, denoted as CRP, which is to minimize the total distribution cost subject to the constraint that the maximum tardiness of the orders, $T_{\max}$, is no more than a given value T_0. We will see later that a lower bound of the optimal objective value of the problem P2 can be obtained by utilizing lower bounds of the optimal objective values of the problem CRP with various values of T_0.

We first formulate CRP as an IP problem. Let Ω_i be the set of all feasible schedules for a single batch of orders from customer i, for $i \in M$. A feasible schedule $\omega \in \Omega_i$ for a batch of orders from customer $i \in M$ specifies which orders are in the batch, the starting time of the first order, the processing completion time of the last order, and the time these orders are delivered to the customer. All feasible schedules for a batch satisfy the constraint that the maximum delivery tardiness of the orders in the batch is no more than the given value T_0. We define the following parameters:

Q = total processing times of all the orders = $\sum_{(i,j)\in N} p_{ij}$

g_ω = transportation cost of schedule $\omega \in \Omega_i = f_i$

$a_{j\omega} = 1$ if order (i,j) is covered in schedule $\omega \in \Omega_i$ and 0 otherwise.

$\tau_{t\omega} = 1$, if time interval $[t, t+1]$ is covered by schedule $\omega \in \Omega_i$ and 0 otherwise.

Also, we define a variable x_ω to be 1 if schedule $\omega \in \Omega_i$ is used and 0 otherwise. Then CRP can be formulated as the following set partitioning type binary IP formulation:

$$[SP] \qquad \min \sum_{i\in M}\sum_{\omega\in\Omega_i} g_\omega x_\omega \tag{3.17}$$

Subject to:

$$\sum_{\omega\in\Omega_i} a_{j\omega}x_\omega = 1 \quad i \in M, j = 1,\ldots,n_i \tag{3.18}$$

$$\sum_{i\in M}\sum_{\omega\in\Omega_i} \tau_{t\omega}x_\omega = 1 \quad t = 0,1,\ldots,Q-1 \tag{3.19}$$

$$x_\omega \in \{0,1\} \qquad \omega \in \cup_{i\in M}\Omega_i \tag{3.20}$$

In [SP], the objective function is to minimize the total distribution cost. Equation (3.18) ensures that each order gets covered exactly once by some schedule. Equation (3.19) ensures that each time slot in the interval $[0, Q]$ is covered exactly once.

We denote the LP relaxation of [SP] as [LSP] where the constraint (3.20) is replaced by "$x_\omega \geq 0$". Clearly, the optimal objective value of [LSP] is a lower bound of that of CRP. We will develop a column generation based algorithm to solve [LSP].

Next we describe how to get a lower bound for problem P2 by solving [LSP]. Let Ψ denote the set of all possible values of $T_{\max}$ in a feasible schedule of problem P2. Since the order processing times and the due dates are integer valued, there is only a finite number of values in the set Ψ. The minimum value of $T_{\max}$, denoted as $T_{\max}^{\min}$, is obtained when the orders are processed in their SEDD order and then shipped individually. The maximum value possible for $T_{\max}$, denoted as $T_{\max}^{\max}$, is given by the maximum tardiness of orders if they are processed in their SEDD order and all are delivered in full batches except possibly last several orders that are delivered in a partial batch. Let $LB_{CRP}(T_0)$ denote the optimal objective value of [LSP] with $T_{\max}$ no more than T_0. Then it can be seen that $LB_{P2} = \min\{\alpha T_0 + (1-\alpha)LB_{CRP}(T_0) | T_0 \in \Psi\}$ is a valid lower bound of P2. However, it may not be necessary to solve [LSP] for each value of T_0 in Ψ.

Next we give a procedure for getting a lower bound for P2 based on the above observations.

Algorithm A4

Step 0: Set the lower bound, $LB = \infty$. Set $T_0 = T_{\max}^{\max}$.

Step 1: Solve [LSP]. Let the optimal objective value be Z^*.

Step 2: Obtain the actual maximum tardiness value corresponding to the current optimal solution of [LSP], denoted as $T_{\max}^a$, which is defined to be the maximum tardiness among all the batches $\omega \in \Omega_i$, $i \in M$ whose corresponding variable in [LSP] has a positive value. If $\alpha T_{\max}^a + (1-\alpha)Z^* < LB$, set $LB = \alpha T_{\max}^a + (1-\alpha)Z^*$.

Step 3: Set $T_0 = T_{\max}^a - 1$. If $T_0 \geq T_{\max}^{\min}$, go to Step 1. Otherwise, STOP and LB gives a lower bound for P2.

It should be noted that due to the way we reduce in Step 3, we need not solve the [LSP] for each and every integer value of $T_{\max}$ between $T_{\max}^{\min}$ and $T_{\max}^{\max}$.

Column generation for solving [LSP]

Due to the large number of columns in the formulation [LSP], it is impractical to solve it directly. We resort to a column generation approach. In each iteration of the column generation approach, we first solve a master problem, which is [LSP] with only a subset of the columns (i.e. single-batch schedules) from each set Ω_i. Then we use the dual variable values of the master problem to form a subproblem corresponding to each customer to find schedules $\omega \in \Omega_i$ with a negative reduced cost. We add to the master problem several columns with negative reduced costs generated while solving these subproblems. Then we solve the master problem again. We repeat this till there are no more columns that can be generated by solving the subproblems that give negative reduced costs. At that point, we have the optimal solution to [LSP].

An initial set of columns for [LSP] can be generated by processing all the orders in the SEDD sequence and delivering them individually. We use ρ_{ij} and γ_t to denote the dual variable value corresponding to the constraint set (3.18) and (3.19) of [LSP], respectively. Then the reduced cost r_ω of a column corresponding to $\omega \in \Omega_i$, $i \in M$, is given by:$r_\omega = g_\omega - \sum_{j\in\omega} a_{j\omega}\rho_{ij} - \sum_{t\in\omega} \tau_{t\omega}\gamma_t$. The ith subproblem (for customer $i \in M$) is to select columns $\omega \in \Omega_i$ with the minimum value of r_ω. Before we present an algorithm for solving this subproblem, we first note that the following are true for each column $\omega \in \Omega_1 \cup \ldots \cup \Omega_m$:

(a) All the orders in the column belong to the same customer

(b) There are no more than b orders in the column

(c) The maximum tardiness for the orders in the column is not greater than the given value

T_0

(d) The completion time of the last order in the column must be no more than Q, the total processing times of all the orders in N.

By Lemma 9, for the special case of the problem P2A, in addition to (a)-(d) above, the orders in a batch will be consecutive orders from the EDD sequence for the customer. For the general case of the problem P2, the orders in a batch of a customer may not be consecutive in the EDD sequence for that customer. Hence the number of potential columns is much higher in the case of P2.

In the following, we develop a dynamic programming algorithm for solving the ith sub-problem, for any $i \in M$. The algorithm is presented for the case of problem P2, and can be simplified slightly for the case of P2A.

Algorithm DP2

Define the value function $F(u, v, q, t)$ as the minimum reduced cost of a schedule of a batch with q orders that contains order (i, u) as the first order and (i, v) as the last order which is completed at time t, where $v \geq u, q \leq b$ and $t \leq \min\{Q, d_{iu} + T_0 - t_i\}$.

Initial values

$$\begin{aligned} F(u,u,1,t) &= f_i - \rho_{iu} - \sum_{h=t-p_{iu}}^{t-1} \gamma_h, \text{ for } u \in \{1,\ldots,n_i\}, t \in \{p_{iu},\ldots,\min\{Q, d_{iu}+T_0-t_i\}\} \\ F(u,u,q,t) &= \infty, \text{ if } q > 1 \text{ or } t \notin \{p_{iu},\ldots,\min\{Q, d_{iu}+T_0-t_i\}\} \end{aligned}$$

Recursive relations

For $u \in \{1, \ldots, n_i\}, v \in \{u, \ldots, n_i\}, q \in \{1, \ldots, b\}$, and $t \in \{p_{iu}, \ldots, \min\{Q, d_{iu} + T_0 - t_i\}\}$,

$$F(u,v,q,t) = \min\{F(u,k,q-1,t-p_{iv}) - \rho_{iv} - \sum_{h=t-p_{iv}}^{t-1} \gamma_h | k = u,\ldots,v-1\} \quad (3.21)$$

Optimal solution

For a fixed $u \in \{1, \dots, n_i\}$, an optimal schedule with order (i, u) as the first order is found by minimizing $F(u, v, q, t)$ over all possible (v, q, t) satisfying: $v \geq u, 1 \leq q \leq b$, and $p_{iu} \leq t \leq \min\{Q, d_{iu} + T_0 - t_i\}$. Among the n_i such schedules found, the one with the minimum F is optimal to subproblem i.

Lemma 15 Algorithm DP2 solves the ith subproblem optimally in time $O(n_i^3 bQ)$.

Proof The term $-\rho_{iv} - \sum_{h=t-p_{iv}}^{t-1} \gamma_h$ in the recursive relation (3.21) is the total contribution to the reduced cost made by order (i, v) which is scheduled in the interval $[t - p_{iv}, t)$. The recursive relation enumerates all possible orders (i, k) that can be scheduled before the current last order (i, v). Thus the optimality is guaranteed. There are a total of $O(n_i^2 bQ)$ states in the dynamic program and it takes $O(n_i)$ time to compute the value for each state, thus the overall time needed by the algorithm is $O(n_i^3 bQ)$. ■

Algorithm DP2 can be made more efficient for the ith subproblem in the case of problem P2A. As we pointed out earlier, for problem P2A, each delivery batch for a customer consists of consecutive orders from the EDD sequence of the orders of that customer. Thus, $q = v - u + 1$ in each state of the dynamic program, which means that we can actually drop q from each state and the recursive relation (3.21) can be modified as:

$$F(u, v, t) = F(u, v - 1, t - p_{iv}) - \rho_{iv} - \sum_{h=t-p_{iv}}^{t-1} \gamma_h$$

The time complexity of the DP becomes $O(n_i bQ)$.

In solving [LSP], some techniques can be used to speed up the algorithm. For example, it is not necessary to run DP2 for every choice of the first order (i, u) in each iteration of the column generation. We use a cyclic scheme for running DP2 in which we start with order $(i, 1)$ as the first order in the first iteration of the column generation and continue with $(i, 2)$ as the first order and so on until we generate a certain number of columns with a negative reduced cost, and then in the next iteration of the column generation we start from the last order considered in the last iteration as the first order in the batch.

Computational results

In this section, we describe the computational experiment to evaluate the performance of the heuristic H1. The results obtained from the heuristic are compared with the lower bound generated using the column generation approach. Test problems are randomly generated as follows:

(a) Total number of orders $n \in \{20, 30, 40\}$; number of customers $m \in \{2, 4\}$; shipment capacity $b \in \{2, 4\}$; The orders are assigned to customers randomly, with each order having an equal probability for getting assigned to a particular customer.

(b) Order processing times p_{ij} are independently generated from a uniform distribution $U[1, 10]$.

(c) Transportation times t_i are independently generated from a uniform distribution $U[10, 100]$; transportation cost per delivery shipment f_i is set equal to the transportation times.

(d) Order due dates d_{ij} are independently generated from a uniform distribution $U[p_{\min} + t_{\min}, \lambda((p_{\min} + p_{\max})n/2 + (t_{\min} + t_{\max})/2)]$, where $p_{\min}$ and $p_{\max}$ are the minimum and maximum order processing times respectively, and $t_{\min}$ and $t_{\max}$ are the minimum and maximum transportation times. Value of λ determines how tight the due dates are. We test three different levels: $\lambda \in \{0.5, 1, 1.5\}$. Two types of due dates are considered: special (corresponding to problem P2A) and general (corresponding to problem P2). For the special case, due dates are made agreeable with the processing times. To ensure this, the same random seed is used to generate both the processing time and the due date for an order. For the general case, the due dates are generated independent of the processing times.

(e) Weighting parameter in the objective function $\alpha \in \{0.5, 0.75, 0.9\}$. The values for α are chosen so that we are able to demonstrate the effect of varying weights on the production and distribution part. The contribution due to the production part is small compared to that of the distribution part when $\alpha = 0.5$, the two are comparable when $\alpha = 0.75$, and when $\alpha = 0.9$, the production part dominates.

For each of the 108 combinations of $(n, m, b, \lambda, \alpha)$, we test ten different randomly generated problem instances. Five of these are with special due dates (i.e. for problem P2A) while the remaining five are assigned general due dates (i.e. for problem P2). Hence we test a total of 1080 problem instances. The programs were written in C and all LP problems involved were solved by calling the LP solver of CPLEX 8.0. The code was run on a PC with a 1.5-GHz Pentium IV processor and 512-MB memory. Every problem instance was successfully solved by the heuristic with no more than 1 CPU second. On the other hand, the computational time for the column generation procedure was observed to increase at an exponential rate. For problem instances with general due dates, it took around 45 CPU minutes per instance when the number of orders was set at 40. Moreover, the computer ran out of memory when the number of orders was increased beyond 40 for shipment capacity 4.

Table 3.1 reports both average and maximum relative gaps between the objective values Z_{H1} of the solutions generated by the heuristic H1 and the lower bound LB_{P2} generated by solving [LSP] by the column generation approach. The relative gap is defined as $\frac{Z_{H1} - LB_{P2}}{LB_{P2}} \times 100\%$. Clearly, the relative gap defined here is an upper bound of the actual relative gap between the heuristics solutions and the optimal solution. Each entry in the columns "Avg Gap" of Table 3.1 is the average relative gap over the random test problems with the corresponding (n, m, b, α) combination. Note that the results corresponding to different values of λ have been put together for ease of presentation. In order to account for this, we have presented the maximum gap values in each category along with the average.

These results demonstrate that the heuristic is capable of generating near optimal solutions for most problems tested. Due to the excessive computational time needed for getting lower bounds by the column generation approach for larger problems, we did not test on larger problems. However, by the asymptotic optimality of the heuristic (Theorem 15), it can be expected that the heuristic will also perform well for larger problems.

Some other conclusions can be made based on the results in Table 3.1. It can be seen that in general, for a given number of orders n, the performance of the heuristic deteriorates as the maximum allowed batch size b increases. This can be explained by the fact that when we increase the maximum allowed batch size b, the schedules $\pi_1, \ldots, \pi_m$ generated from individual customers in Step 2 are more likely to overlap with one another and hence the final combined schedule p generated in Step 3 is more different from these individual schedules, which leads to a negative effect on the performance of the heuristic. The heuristic performs considerably better when the due dates are proportional to the processing times. This is expected since this heuristic is developed based on this special case. In general, when the number of orders n is high compared to the number of customers m or the maximum allowed batch size b, the heuristic is more likely to generate near optimal solutions.

3.4 Value of Production-Distribution Integration

The problems we have studied integrate order processing and order delivery decisions in order to optimize a combined objective function. However, production and distribution decisions are often treated separately and sequentially in the literature. Most production scheduling models consider order processing only, whereas most distribution models assume that orders to be delivered have been processed and are only concerned with the total distribution cost.

In this section, we analyze the value of such integration. We compare the integrated scheduling approach considered in this chapter with two typical sequential approaches that treat order processing and order delivery sequentially with no or only partial integration. In both the sequential approaches, the production part assumes that each order $(i, j) \in N$ once completed processing is delivered to its customer immediately without considering the possibility of delivery consolidation with other orders, i.e. $D_{ij} = C_{ij} + t_i$, and tries to minimize the maximum tardiness of orders $T_{\max}$. Clearly, scheduling the orders in the SEDD order is optimal in this part. The distribution part of the first sequential approach tries to minimize the distribution cost G only, given the SEDD processing sequence of orders. As a result, the orders completed in the production part are delivered to the customers using a minimum possible number of shipments. Thus, for each customer $i \in M$, the orders $(i, (k-1)b+1), \ldots, (i, kb)$ are delivered together as the kth shipment for $k = 1, \ldots, \lfloor n_i/b \rfloor$, and the remaining orders as the last shipment. The total overall cost $\alpha T_{\max} + (1-\alpha)G$ of this approach can be calculated accordingly. In this sequential approach, production and distribution are treated totally separately and there is no integration at all.

In the second sequential approach, given the SEDD processing sequence of the orders, the distribution part tries to minimize the integrated objective function $\alpha T_{\max} + (1-\alpha)G$. Since the production part does not consider this overall objective, production and distribution is only partially integrated in this sequential approach. In the distribution part of this approach, an optimal distribution schedule can be obtained by applying the first two steps of heuristic H1 to the given SEDD production sequence of the orders with the following two modifications: (i) in the single-customer auxiliary problem AUX_i for customer $i \in M$, the departure time of a delivery batch B is redefined simply as the completion time C_{ij}^{SEDD} of the last order (i,j) in B; (ii) the recursive relation (3.3) of DP1 is replaced by the following:

$$F(j,k) = \min_{1 \leq q \leq \min\{b,j\}} \left\{ \max\left\{ F(j-q,k-1), \max\left\{0, C_{ij}^{SEDD} + t_i - d_{i,j-q+1}\right\}\right\}\right\}$$

Then the total cost of this approach is given by (3.4).

We conduct a computational experiment to evaluate the possible improvement that can be achieved for the integrated objective function, $\alpha T_{\max} + (1-\alpha)G$, from the two sequential approaches to the integrated approach. More specifically, we calculate the relative gap of the objective value of the solution generated by a sequential approach and that generated by the heuristic H1: $\frac{Z_{SEQ} - Z_{H1}}{Z_{SEQ}} \times 100\%$, where Z_{SEQ} and Z_{H1} represent the objective values of the solutions found by a sequential approach and the heuristic H1 respectively. Since the heuristic solution is used instead of the optimal solution for the integrated approach, this relative gap is a lower bound of the relative gap between the sequential approaches and the optimal integrated approach. This gap gives an indication of the percentage savings that we can obtain by resorting to an integrated approach.

Test problems are generated exactly the same way as in Section 3.3.3 except that the number of orders $n \in \{25, 50, 100\}$. Tables 3.2 and 3.3 report the average and maximum gap values for the test problems between the two sequential approaches and the integrated approach. Over all the test problems, the average gap between the first sequential approach and the integrated approach is 6.08% for P2A and 7.35% for P2, whereas that between the second sequential approach and the integrated approach is 1.82% for P2A and 2.20% for P2. This means that the second sequential approach (with partial integration) provides much closer solutions to optimal solutions than the first sequential approach (without any integration). This shows that even just partial integration enhances overall solutions significantly. The results show that the gaps could be as high as 72.32% for the first sequential approach and 21.22% for the second. We also note that since the heuristic is not guaranteed to give the optimal solution for the integrated problem, theoretically it is possible for the sequential approach to beat the heuristic. But as the results show, this does not happen very often. Out of 1080 instances tested, the first sequential approach beat the heuristic in just four instances. For the second sequential approach, this happened 48 times.

Figure 3.2: Average gap for the two sequential approaches

We can also see that in both tables, the gap increases in direct proportion to the maximum allowed batch size b and the number of customers m. This is expected since the effect due to batching becomes more prominent when the number of customers or the maximum allowed batch size is increased. It may also be noted that the value of α plays an important role. At low values of α, the integrated approach is not significantly better even compared to the first sequential approach since we are laying emphasis on the distribution cost and the first sequential approach minimizes this. But as the value of α increases, we see a significant increase in the gap. When everything else is kept constant, increasing the number of orders n leads to a decrease in the gap for both the sequential approaches. This is shown in Fig 3.2. This is explained by the fact that the heuristic H1 is essentially a local perturbation around the SEDD sequence. When the number of orders is increased, the change in tardiness value through this local perturbation does not increase proportionately. Hence in general, when the number of orders is very high compared to the maximum allowed batch size or the number of customers, the performance of heuristic H1 is not significantly better than that of the sequential approach. The integrated approach leads to good improvements in performance under cases where the contribution due to the maximum tardiness is significant in the objective function value.

3.5 Conclusions

In this chapter, we have studied the production-distribution system with one supplier and one or more customers. Our goal was to optimize a combined objective function that considered both the maximum tardiness and total distribution cost. It was seen that for an arbitrary number of customers, the problem is NP-hard even in the special case where the processing times and the due dates are agreeable. A fast heuristic has been proposed that is asymptotically optimal when the number of orders goes to infinity. Computational tests show that the heuristic is capable of generating near optimal solutions. We have also demonstrated that there is distinct advantage of using the integrated production-distribution approach as compared to the two sequential approaches that try to optimize production and delivery sequentially with no or only partial integration.

It should be noted that though we have assumed b, the maximum allowed batch size, to be the same for every customer, it is not difficult to extend the heuristic and all the other algorithms to the case where the maximum allowed batch size is dependent on the customers. All the results presented in the chapter still hold. We have shown that in the case when the processing times and due dates are agreeable, there exists a procedure that is polynomial in the number of orders that can solve the problem optimally. The complexity of the case with general processing times and due dates and a fixed number of customers is left as an open problem.

In the case when there is no batch size limit, i.e. $b = n$, the problems P2A and P2 with an arbitrary number of customers are still NP-hard because the same NP-hardness proof given in Section 3.2.1 still works for this case. On the other hand, when $b = n$, both problems P1 and P2 with a fixed number of customers can be solved in polynomial time by the $O(n^{2m+1})$ dynamic programming algorithm of Hall and Potts (2003) mentioned in Section 3.2.1 after it is slightly modified to take transportation times ti into account. The $O(n^3(\log n)(\log(P + t)))$ algorithm given in Section 3.2.2 still works for problem P1 with $n = b$. However, the algorithm of Hall and Potts has a lower time complexity.

In this chapter, we have not considered shipments that can serve more than one customer. Such a problem would include routing decisions for each shipment. When sharing of shipments across different customers is allowed, the shortest route may not always be the best since we have to take into account the tardiness for the orders delivered at each customer. Consequently, we cannot define the shipping due dates any more as the orders of a customer may be routed through some other customers before getting delivered. New algorithms would be needed to solve such a problem. We believe that the value of production-distribution integration would be even greater in this case because it would require a closer production-distribution linkage in order to fully take advantage of order consolidation across different customers.

Table 3.1: Computational Results of Heuristic H1

Problem				P2A		P2		Overall	
n	m	b	α	Avg Gap	Max Gap	Avg Gap	Max Gap	Avg Gap	Max Gap
			0.5	0.00%	0.00%	3.58%	8.97%	1.79%	8.97%
20	2	2	0.75	0.00%	0.00%	3.10%	7.09%	1.55%	7.09%
			0.9	0.11%	1.66%	1.88%	4.26%	0.99%	4.26%
			0.5	0.09%	1.28%	9.49%	17.97%	4.79%	17.97%
20	2	4	0.75	0.37%	3.06%	6.61%	15.13%	3.49%	15.13%
			0.9	0.74%	4.61%	4.19%	9.25%	2.46%	9.25%
			0.5	0.03%	0.41%	9.15%	21.18%	4.59%	21.18%
20	4	2	0.75	0.07%	1.03%	6.88%	17.48%	3.47%	17.48%
			0.9	0.53%	6.00%	4.83%	11.47%	2.68%	11.47%
			0.5	0.18%	1.29%	9.65%	14.29%	4.91%	14.29%
20	4	4	0.75	0.39%	2.75%	3.47%	9.04%	1.93%	9.04%
			0.9	1.18%	4.96%	2.50%	7.79%	1.84%	7.79%
			0.5	0.00%	0.00%	5.90%	6.34%	2.95%	6.34%
30	2	2	0.75	0.00%	0.00%	4.88%	6.21%	2.44%	6.21%
			0.9	0.00%	0.00%	3.20%	6.00%	1.60%	6.00%
			0.5	0.15%	1.42%	9.53%	17.07%	4.84%	17.07%
30	2	4	0.75	0.37%	3.61%	7.03%	12.96%	3.70%	12.96%
			0.9	0.98%	4.93%	5.14%	19.01%	3.06%	19.01%
			0.5	0.00%	0.00%	4.14%	6.09%	2.07%	6.09%
30	4	2	0.75	0.00%	0.00%	3.74%	6.75%	1.87%	6.75%
			0.9	0.20%	1.28%	3.86%	7.94%	2.03%	7.94%
			0.5	0.42%	2.11%	11.84%	18.66%	6.13%	18.66%
30	4	4	0.75	1.29%	5.26%	7.58%	11.23%	4.44%	11.23%
			0.9	2.45%	10.75%	5.94%	11.26%	4.19%	11.26%
			0.5	0.00%	0.00%	0.18%	0.90%	0.09%	0.90%
40	2	2	0.75	0.00%	0.00%	0.56%	2.57%	0.28%	2.57%
			0.9	0.00%	0.00%	1.30%	6.52%	0.65%	6.52%
			0.5	0.27%	2.04%	2.67%	10.02%	1.47%	10.02%
40	2	4	0.75	0.62%	5.41%	3.72%	9.35%	2.17%	9.35%
			0.9	1.59%	14.58%	4.63%	12.08%	3.11%	14.58%
			0.5	0.03%	0.32%	5.54%	7.30%	2.78%	7.30%
40	4	2	0.75	0.08%	0.89%	4.88%	6.16%	2.48%	6.16%
			0.9	0.61%	3.91%	4.08%	6.86%	2.34%	6.86%
			0.5	0.44%	3.05%	11.74%	16.69%	6.09%	16.69%
40	4	4	0.75	1.64%	6.50%	8.40%	13.22%	5.02%	13.22%
			0.9	3.69%	19.94%	5.89%	8.61%	4.79%	19.94%

Table 3.2: Relative improvement from the first sequential approach to the integrated approach

Problem				P2A		P2		Overall	
n	m	b	α	Avg Gap	Max Gap	Avg Gap	Max Gap	Avg Gap	Max Gap
			0.5	0.55%	2.35%	0.56%	2.63%	0.56%	2.63%
25	2	2	0.75	1.45%	6.36%	1.46%	7.33%	1.45%	7.33%
			0.9	3.26%	14.80%	3.37%	18.18%	3.32%	18.18%
			0.5	2.56%	7.56%	3.15%	8.92%	2.86%	8.92%
25	2	4	0.75	5.70%	16.79%	7.09%	18.91%	6.40%	18.91%
			0.9	10.97%	37.56%	13.38%	30.19%	12.17%	37.56%
			0.5	4.41%	17.43%	3.86%	8.06%	4.14%	17.43%
25	4	2	0.75	9.94%	36.49%	8.99%	18.91%	9.46%	36.49%
			0.9	17.37%	57.42%	16.31%	34.28%	16.84%	57.42%
			0.5	10.40%	31.35%	10.05%	15.19%	10.23%	31.35%
25	4	4	0.75	19.67%	53.02%	19.59%	29.88%	19.63%	53.02%
			0.9	30.46%	72.32%	30.62%	47.79%	30.54%	72.32%
			0.5	0.16%	0.90%	0.37%	2.92%	0.27%	2.92%
50	2	2	0.75	0.46%	2.60%	0.99%	7.72%	0.72%	7.72%
			0.9	1.22%	6.90%	2.39%	17.05%	1.80%	17.05%
			0.5	0.84%	3.29%	1.75%	5.26%	1.29%	5.26%
50	2	4	0.75	2.38%	8.93%	4.32%	12.38%	3.35%	12.38%
			0.9	5.49%	20.83%	9.90%	25.54%	7.70%	25.54%
			0.5	0.73%	5.32%	2.18%	8.15%	1.45%	8.15%
50	4	2	0.75	1.94%	13.85%	5.38%	19.57%	3.66%	19.57%
			0.9	4.46%	29.84%	10.66%	36.72%	7.56%	36.72%
			0.5	6.23%	19.03%	7.30%	15.53%	6.77%	19.03%
50	4	4	0.75	13.41%	37.20%	15.16%	31.73%	14.28%	37.20%
			0.9	23.81%	63.20%	25.10%	52.02%	24.46%	63.20%
			0.5	0.26%	1.62%	0.18%	0.63%	0.22%	1.62%
100	2	2	0.75	0.71%	4.42%	0.51%	1.79%	0.61%	4.42%
			0.9	1.69%	10.46%	1.49%	4.79%	1.59%	10.46%
			0.5	0.58%	3.27%	0.96%	2.99%	0.77%	3.27%
100	2	4	0.75	1.56%	8.26%	2.38%	7.63%	1.97%	8.26%
			0.9	4.04%	15.61%	6.01%	15.83%	5.03%	15.83%
			0.5	0.93%	4.04%	1.30%	2.73%	1.11%	4.04%
100	4	2	0.75	2.44%	10.81%	3.47%	7.40%	2.96%	10.81%
			0.9	5.54%	24.48%	8.03%	17.18%	6.78%	24.48%
			0.5	2.54%	5.83%	4.95%	10.72%	3.75%	10.72%
100	4	4	0.75	6.39%	15.25%	11.32%	23.14%	8.85%	23.14%
			0.9	14.49%	34.47%	20.10%	41.97%	17.29%	41.97%

Table 3.3: Relative improvement from the second sequential approach to the integrated approach

Problem				P2A		P2		Overall	
n	m	b	α	Avg Gap	Max Gap	Avg Gap	Max Gap	Avg Gap	Max Gap
			0.5	0.34%	2.35%	0.33%	2.17%	0.33%	2.35%
25	2	2	0.75	0.72%	4.57%	0.87%	5.71%	0.79%	5.71%
			0.9	0.38%	2.16%	1.40%	7.80%	0.89%	7.80%
			0.5	1.45%	7.56%	1.23%	3.80%	1.34%	7.56%
25	2	4	0.75	1.42%	10.66%	2.38%	5.73%	1.90%	10.66%
			0.9	1.14%	6.69%	0.64%	4.21%	0.89%	6.69%
			0.5	3.21%	7.22%	1.23%	3.96%	2.22%	7.22%
25	4	2	0.75	3.96%	7.58%	2.41%	8.82%	3.18%	8.82%
			0.9	3.60%	9.18%	2.23%	9.70%	2.92%	9.70%
			0.5	6.10%	21.22%	7.07%	15.19%	6.59%	21.22%
25	4	4	0.75	8.74%	19.85%	8.78%	20.33%	8.76%	20.33%
			0.9	8.21%	18.60%	5.20%	13.78%	6.71%	18.60%
			0.5	0.07%	0.50%	0.35%	2.92%	0.21%	2.92%
50	2	2	0.75	0.18%	1.39%	0.78%	5.64%	0.48%	5.64%
			0.9	0.47%	3.46%	0.93%	4.52%	0.70%	4.52%
			0.5	0.42%	2.65%	1.11%	3.87%	0.76%	3.87%
50	2	4	0.75	0.93%	4.38%	1.31%	4.09%	1.12%	4.38%
			0.9	0.88%	6.54%	0.99%	7.70%	0.94%	7.70%
			0.5	0.56%	3.96%	1.18%	3.50%	0.87%	3.96%
50	4	2	0.75	1.02%	4.39%	2.86%	8.81%	1.94%	8.81%
			0.9	1.33%	4.25%	3.06%	7.05%	2.19%	7.05%
			0.5	3.88%	10.32%	3.79%	8.72%	3.83%	10.32%
50	4	4	0.75	5.17%	13.21%	5.86%	12.05%	5.51%	13.21%
			0.9	4.55%	11.77%	5.47%	11.93%	5.01%	11.93%
			0.5	0.04%	0.43%	0.13%	0.59%	0.08%	0.59%
100	2	2	0.75	0.12%	1.22%	0.30%	1.69%	0.21%	1.69%
			0.9	0.09%	1.92%	0.43%	1.92%	0.26%	1.92%
			0.5	0.37%	2.19%	0.41%	1.25%	0.39%	2.19%
100	2	4	0.75	0.45%	2.68%	0.81%	2.50%	0.63%	2.68%
			0.9	0.41%	3.07%	1.20%	4.20%	0.81%	4.20%
			0.5	0.57%	2.19%	0.93%	2.53%	0.75%	2.53%
100	4	2	0.75	0.73%	2.84%	2.16%	5.27%	1.45%	5.27%
			0.9	0.30%	2.72%	2.29%	5.49%	1.29%	5.49%
			0.5	1.46%	4.32%	2.08%	5.66%	1.77%	5.66%
100	4	4	0.75	1.78%	6.35%	3.71%	9.11%	2.75%	9.11%
			0.9	0.65%	8.35%	3.43%	9.34%	2.04%	9.34%

Chapter 4

Joint Cyclic Production and Delivery Scheduling in a Two-Stage Supply Chain

4.1 Introduction

Production and distribution operations are the two most important operational functions in a supply chain. It is critical to plan and schedule these two functions in a coordinated manner in order to achieve optimal operational performance of the supply chain. In this chapter, we study an integrated production and distribution scheduling model in a two-stage supply chain consisting of one or more suppliers, a warehouse, and a customer. Each supplier manufactures a unique item at a constant rate. The customer's demand for each item is constant and known in advance. Each supplier manufactures its item in batches and there is a setup time and setup cost incurred for every production batch. Manufactured items are shipped directly from the suppliers to the warehouse, and from the warehouse to the customer. In the delivery from the warehouse to the customer, different products from the suppliers are consolidated and shipped together. There are inventory holding costs at all the facilities (suppliers, warehouse, and customer) and there are transportation costs for deliveries from the suppliers to the warehouse and from the warehouse to the customer. The objective is to find a joint cyclic production and delivery schedule over an infinite planning horizon to minimize the total production, inventory and transportation cost per unit time.

Figure 4.1 illustrates the supply chain we consider.

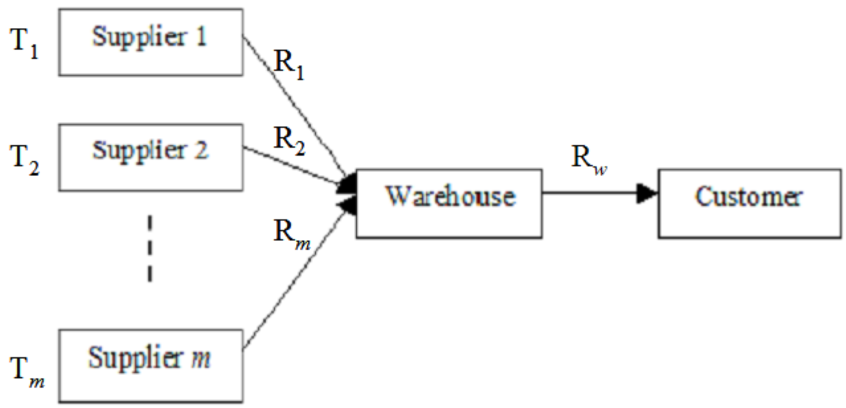

Figure 4.1: A two-stage supply chain

We define the following notation:

m : number of suppliers.

i : supplier and product index, $i = 1, \ldots, m$. (Supplier i produces product i)

D_i : customer demand rate for product i, for $i = 1, \ldots, m$.

p_i : unit processing time of product i, for $i = 1, \ldots, m$.

s_i, S_i : setup time and setup cost per production batch at supplier i respectively, for $i = 1, \ldots, m$.

h_{si}, h_{wi}, h_{ci} : unit inventory holding cost for product i at supplier i, at the warehouse, and at the customer, respectively, for $i = 1, \ldots, m$.

A_i : transportation cost per delivery from supplier i to the warehouse, for $i = 1, \ldots, m$.

A_w : transportation cost per delivery from the warehouse to the customer.

For ease of presentation, we assume that both the delivery time from a supplier to the warehouse and that from the warehouse to the customer are negligible, and hence they are set to zero. This assumption can be easily relaxed. At each supplier, we have $p_i D_i \leq 1$ in order to satisfy the customer demand subject to the capacity constraint (the inequality is strict unless the setup time is zero). We also assume that the unit inventory holding cost of a product at the customer is the highest whereas that at the supplier is the lowest, i.e. $h_{si} \leq h_{wi} \leq h_{ci}$, for $i = 1, \ldots, m$. This assumption reflects the situation in many supply chains where the customers (e.g. retailers) are located at the most populated areas and hence have the highest unit inventory holding cost because of the tight space limit, whereas the suppliers (e.g. plants) are located at places with very low holding costs.

Given these parameters, we need to find a joint cyclic production and delivery schedule, which is equivalent to finding the following cycle times and the relative positions of these cycles:

T_i : time between successive production setups at supplier i, for $i = 1, \ldots, m$.

R_i : time between successive deliveries from supplier i to the warehouse, for $i = 1, \ldots, m$.

R_w : time between successive deliveries from the warehouse to the customer.

Since the schedules are cyclic, in each production or delivery cycle, exactly the same amount will be produced or delivered. Consequently, for $i = 1, \ldots, m$, in each production cycle at supplier i, T_iD_i units of product i need to be produced, and in each delivery cycle from supplier i to the warehouse, R_iD_i units of product i need to be delivered. Clearly, there is only one product involved in a production batch at each supplier and in a delivery from a supplier to the warehouse. However, all the m products are included in a delivery from the warehouse to the customer. That is, in each delivery cycle from the warehouse to the customer, R_wD_1 units of product 1, R_wD_2 units of product 2, ..., and R_wD_m units of product m need to be delivered.

We consider the following two policies for production and delivery cycles:

i) Production cycle time at each supplier is the same as the delivery cycle time from that supplier to the warehouse, i.e. $T_i = R_i$, for $i = 1, \ldots, m$.

ii) Production cycle time at each supplier is an integer multiple of the delivery cycle time from that supplier to the warehouse, and the delivery cycle time from a supplier to the warehouse is an integer multiple of the delivery cycle time from the warehouse to the customer, i.e. $T_i = M_{si} R_i$ and $R_i = M_{wi} R_w$, for $i = 1, \ldots, m$, where $M_{s1}, \ldots, M_{sm}$ and $M_{w1}, \ldots, M_{wm}$ are all positive integers.

These policies are similar to some commonly considered policies for similar models in the literature. Our consideration of these policies is partially justified by the following result.

Lemma 16 In an optimal cyclic schedule to our model, $T_i \geq R_i \geq R_w$, for $i = 1, \ldots, m$.

Proof We prove this by contradiction. In a given cyclic schedule, if $R_i < R_w$ for some supplier i, we modify this schedule by increasing R_i such that $R_i = R_w$. Delivery from supplier i to the warehouse is less frequent in the modified schedule, which brings down the per-period transportation cost. The inventory cost will also decrease or remain the same. This is because $h_{si} \leq h_{wi}$ and when we set $R_i = R_w$, the products wait at the suppliers instead of the warehouse. Hence the modified schedule has a lower total cost. Similarly, if $T_i < R_i$ for some supplier i in a given schedule, we modify this schedule by increasing T_i such that $T_i = R_i$. It can be seen that in this modified schedule both the total production setup cost and the total inventory cost at supplier i are lower than before. The inventory costs go down since in the modified schedule, the products for a delivery batch get processed continuously, with just one setup in the beginning and without a break for setups in between. So on an average, products spend less time waiting at the suppliers. Hence the modified schedule has a lower total cost. ■

This result implies that there is no need to consider policies that either require the delivery cycle time from the warehouse to the customer to be greater than that from a supplier to the warehouse or require the delivery cycle time from a supplier to the warehouse to be greater than the production cycle time at that supplier. However, all the schedules that satisfy policy (i) or (ii) are only a subset of the schedules that satisfy Lemma 1. Hence an optimal cyclic schedule that satisfies policy (i) or (ii) may not be optimal to our model. On the other hand, schedules that satisfy these policies are easier to implement in practice than those that satisfy Lemma 1 but not these policies.

The remainder of this chapter is organized as follows. We then study our model under policy (i) and that under policy (ii) in Sections 4.2 and 4.3, respectively. In Section 4.2, we will show that in an optimal cyclic schedule under policy (i), the delivery cycle time from each supplier to the warehouse is an integer multiple of the delivery cycle time from the warehouse to the customer, i.e. $R_i = M_{wi}R_w$, for $i = 1, \ldots, m$ and some positive integers $M_{w1}, \ldots, M_{wm}$. We will show that for the model with a single supplier, an optimal cyclic schedule can be obtained by closed-form formulas. For the model with multiple suppliers, we propose a heuristic and evaluate the performance of the heuristic computationally. In Section 4.3, we propose and computationally evaluate a heuristic for the problem under policy (ii) with multiple suppliers. We then evaluate the value of warehouse in our two-stage supply chain by comparing this supply chain with a single-stage supply chain without the warehouse. The total cost obtained through our heuristic for the two-stage model under policy (ii) is compared to the total cost obtained by an optimal algorithm from the literature for the single-stage model without the warehouse. Managerial insights are derived based on an extensive set of computational tests. It is conceptually well-understood that a warehouse plays an important role in a supply chain; it consolidates different products and positions the inventory closer to customers, and hence saves on transportation and inventory costs. Our study here attempts to quantify these benefits for the supply chain we consider. Finally, we conclude the chapter in Section 4.4.

4.2 The Model under Policy (i)

We first prove an optimality property and derive the various cost components for the model under policy (i) in Section 4.2.1. We then give an optimal solution to the case with a single supplier in Section 4.2.2, and propose a heuristic for the case with multiple suppliers and evaluate its performance in Section 4.2.3.

4.2.1 An Optimality Property

Theorem 16 In an optimal cyclic schedule under policy (i), the delivery cycle time from each supplier to the warehouse is an integer multiple of the delivery cycle time from the warehouse to the customer, i.e. $R_i = M_{wi} R_w$ for some positive integer M_{wi}, for $i = 1, \ldots, m$.

Proof We prove the theorem for the single-supplier case first and then extend the result to the case with multiple suppliers. When we have only one supplier, we discard the supplier subscript in our notation. Hence the production cycle time and the delivery cycle time from the supplier to the warehouse are denoted as T and R respectively. Under policy (i), we have $T = R$. Suppose that $M_w = R/R_w$ is not an integer. We show that the total cost of a new schedule where both the production cycle time at the supplier and the delivery cycle time from the supplier to the warehouse are increased to $\lceil M_w \rceil R_w$ is not greater than that of the original schedule. Before giving a formal analysis, we explain through a diagram the various categories of inventory at the warehouse when M_w is not an integer. The inventory level over time at the warehouse repeats every kR units of time, where k is the smallest integer such that kR/R_w is an integer. The period of kR time units is hence called an inventory cycle) of the warehouse. For illustration purposes, let us assume $M_w = 2\frac{1}{3}$. Figure 4.2 shows the inventory level at the warehouse over one entire inventory cycle, i.e. over $3R$ time units ($k = 3$ here).

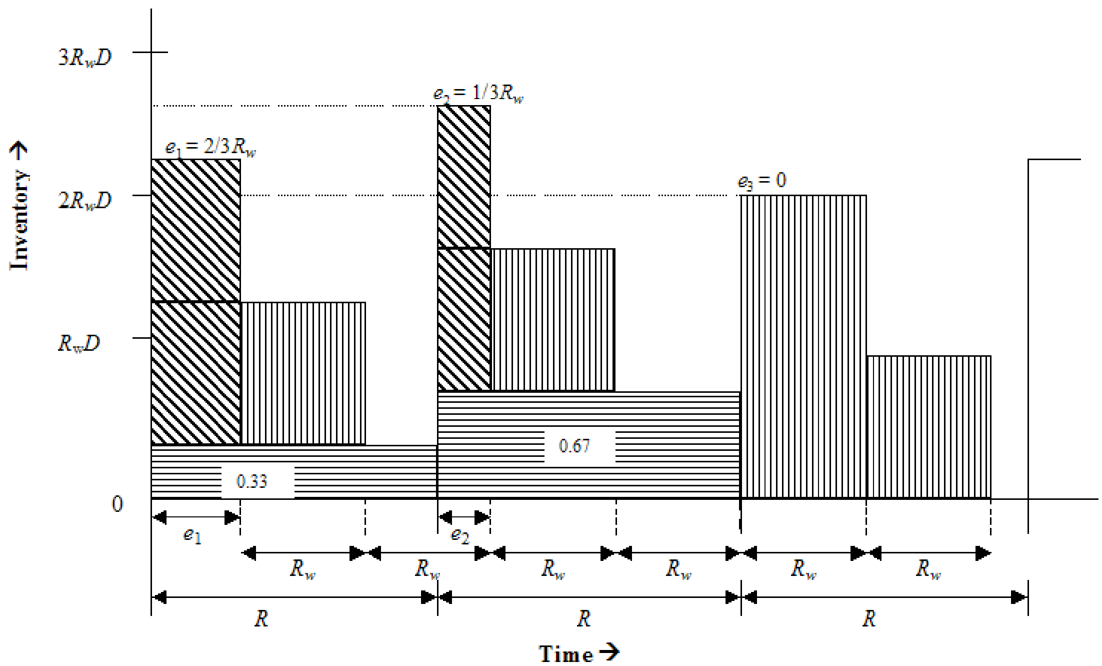

Figure 4.2: Inventory level at the warehouse over one inventory cycle when $M_w = 2\frac{1}{3}$

In Figure 4.2, the solid vertical lines below the horizontal axis (i.e. at times 0, R, $2R$, $3R$) indicate deliveries from the supplier to the warehouse while the dotted lines (i.e. at times e_1, $e_1 + R_w$, $e_1 + 2R_w$, ...) indicate deliveries from the warehouse to the customer. The earliness parameter e_i, for $i \in \{1, 2, 3\}$ represents the gap between the time of the ith delivery from the supplier to the warehouse (i.e. time $(i-1)R$) and the time of the first delivery from the warehouse to the customer after time $(i-1)R$. For the third shipment from the supplier, the delivery from the supplier to the warehouse coincides with a delivery from the warehouse to the customer. Hence $e_3 = 0$. Note that the first delivery from the warehouse to the customer does not take place until time $e_1 = \frac{2}{3}R_w$. Without this intentional delay, the warehouse will not have sufficient inventory on time for some of the future deliveries.

We divide the inventory into three categories: (i) Fractional inventory represented by the areas with the horizontal line shading in Figure 4.2. This is the inventory that has to wait till the next shipment from the supplier before getting delivered to the customer. This inventory is always a fraction of $R_w D$, the demand corresponding to one delivery period from the warehouse to the customer. (ii) Earliness inventory represented by the areas shaded with lines sloping downwards. This corresponds to the earliness e_1, e_2, and e_3 described earlier. (iii) Integral inventory represented by all the remaining areas with a vertical shading. We note that if R/R_w was an integer, this would be the only kind of inventory at the warehouse, as there would be no fractional or earliness inventory.

Now we look at the general case of non-integer M_w. Let $\delta = M_w - \lfloor M_w \rfloor$. Clearly, $0 < \delta < 1$. We first calculate the minimum value of earliness e_1, as minimizing this minimizes the total inventory cost at the warehouse. Let us assume $e_1 = \lambda R_w$ for some $\lambda \in (0, 1)$. Let k be the smallest integer such that kM_w is an integer. We can express δ as (a/k), where $a = kM_w - k\lfloor M_w \rfloor$ is an integer, $a < k$, and a and k are relatively prime to each other. As shown in Figure 4.2, deliveries from the supplier occur at time points $iR = iM_w R_w$, where $i \in \{0, 1, \ldots, k-1\}$. The first delivery to the customer from the warehouse containing orders from shipment $(i + 1)$ from the supplier occurs at time $(\lfloor iM_w \rfloor + \lambda)R_w$. In order that the shipment from the supplier has been delivered by this time, we should have $(\lfloor iM_w \rfloor + \lambda)R_w \geq iR = iM_w R_w$, for $i = 0, \ldots, k-1$. The difference between these two numbers is the earliness for delivery $(i + 1)$ from the supplier, denoted as $e_{(i+1)}$. This implies that $e_{(i+1)} = (\lfloor iM_w \rfloor + \lambda)R_w - iM_w R_w = \left(\lfloor \frac{ia}{k} \rfloor + \lambda - \frac{ia}{k}\right) R_w \geq 0$, or $\lambda \geq ia/k - \lfloor ia/k \rfloor$, for $i = 0, \ldots, k-1$.

We show in the following that the minimum value of λ that satisfies the above condition is $(k-1)/k$. For every $i \in \{0, \ldots, k-1\}$, we can express $\frac{ia}{k}$ as: $\frac{ia}{k} = \lfloor \frac{ia}{k} \rfloor + \frac{r}{k}$, $r \in \{0, \ldots, k-1\}$. We argue that $\frac{ia}{k}$ has a unique remainder $\frac{r}{k}$, for every $i = 0, \ldots, k-1$. Suppose that there exist i and j with $0 \leq i < j \leq k-1$ such that the remainders of $\frac{ia}{k}$ and $\frac{ja}{k}$ are identical.

Then we have:

$$\frac{ja}{k} - \frac{ia}{k} = \frac{(j-i)a}{k} = \left\lfloor \frac{ja}{k} \right\rfloor - \left\lfloor \frac{ia}{k} \right\rfloor \tag{4.1}$$

Since the difference of two integers is an integer, equation (4.1) implies that $(j-i)a$ is an integer multiple of k. This is not possible since $(j-i) < k$, and a and k are relatively prime to each other. Hence, each value of $i \in \{0, \dots, k-1\}$ corresponds to a unique remainder. This implies that there is exactly one i with a remainder value of zero, one with $1/k$, one with $2/k$, and so on. Hence the maximum remainder is $(k-1)/k$. This means that the minimum possible value of λ is $(k-1)/k$.

We now look at the three categories of inventory over the entire inventory cycle of kR time units at the warehouse. Over each interval of R time units, the number of deliveries to the customer is either $\lfloor M_w \rfloor$ or $(\lfloor M_w \rfloor + 1)$. Hence the earliness inventory corresponding to the $(i+1)$th delivery from the supplier to the warehouse is a rectangle with a height of either $\lfloor M_w \rfloor R_w D$ or $(\lfloor M_w \rfloor + 1) R_w D$ and width of $e_{(i+1)} = \left(\lfloor \frac{ia}{k} \rfloor + \lambda - \frac{ia}{k} \right) R_w$, for $i \in \{0, 1, \dots, k-1\}$. The total time in a cycle is $kR = kM_w R_w$ units. Therefore, the average earliness inventory per unit time I_e is:

$$\begin{aligned}
I_e \geq \frac{\sum_{i=0}^{k-1} \left(\lfloor \frac{ia}{k} \rfloor + \lambda - \frac{ia}{k} \right) R_w \lfloor M_w \rfloor R_w D}{k M_w R_w} &= \left(k\lambda + \sum_{i=0}^{k-1} \left(\left\lfloor \frac{ia}{k} \right\rfloor - \frac{ia}{k} \right) \right) \frac{\lfloor M_w \rfloor R_w D}{k M_w} \\
&= \left(k\lambda - \frac{k-1}{2} \right) \frac{\lfloor M_w \rfloor R_w D}{k M_w} \\
&\geq \frac{(k-1) \lfloor M_w \rfloor R_w D}{2 k M_w}
\end{aligned} \tag{4.2}$$

where the last inequality is obtained by letting $\lambda = (k-1)/k$. The fractional inventory level during period $[iR, (i+1)R)$ can be calculated as the difference between the cumulative quantity delivered to the warehouse and the cumulative quantity delivered out of the warehouse by the end of the period. This is given by:

$$(i+1) M_w R_w D - \lfloor (i+1) M_w \rfloor R_w D = \left(\frac{(i+1)a}{k} - \left\lfloor \frac{(i+1)a}{k} \right\rfloor \right) R_w D \tag{4.3}$$

Each period $[iR, (i+1)R)$ is for a duration of $R = M_w R_w$ time units, and each inventory cycle is for a duration of $kM_w R_w$ time units. Therefore, the average fractional inventory per unit time during a cycle of kR time units is given by:

$$I_f = \frac{\sum_{i=0}^{k-1} \left(\frac{(i+1)a}{k} - \left\lfloor \frac{(i+1)a}{k} \right\rfloor \right) R_w D (M_w R_w)}{kM_w R_w} = \frac{(\frac{k-1}{2}) M_w R_w^2 D}{kM_w R_w} = \frac{(k-1) R_w D}{2k} \tag{4.4}$$

where we have used the fact observed earlier that there is a distinct remainder of ia/k for each $i = 0, \ldots, k-1$, which implies that $\sum_{i=0}^{k-1}((i+1)a/k - \lfloor (i+1)a/k \rfloor) = \sum_{i=0}^{k-1} i/k = (k-1)/2$.

Next we calculate the remaining part of the inventory, the integral inventory. If $e_{(i+1)} < (a/k)R_w$ for some $i \in \{0, \ldots, k-1\}$, there will be $\lceil M_w \rceil$ deliveries from the warehouse to the customer during the period $[iR, (i+1)R)$. Otherwise, there will be $(\lceil M_w \rceil - 1)$ deliveries. The first of these $\lceil M_w \rceil$ or $(\lceil M_w \rceil - 1)$ deliveries from the warehouse has already been counted in the form of earliness inventory (See Figure 4.2). Hence the integral inventory begins at a level of $\lfloor M_w \rfloor R_w D$ or $(\lfloor M_w \rfloor - 1) R_w D$ depending on the value of $e_{(i+1)}$. And it reduces by $R_w D$ every R_w time units, finally reaching zero. Based on our analysis following equation (4.1), there exists an $i \in \{0 \ldots k-1\}$, for which $\frac{ia}{k} - \left\lfloor \frac{ia}{k} \right\rfloor = \frac{k-1}{k} = \lambda$. For this value of i, $e_{(i+1)} = \left(\left\lfloor \frac{ia}{k} \right\rfloor + \lambda - \frac{ia}{k} \right) R_w = 0 < (a/k)R_w$. Therefore we have at least one instance where the integral inventory begins at a higher level of $\lfloor M_w \rfloor R_w D$. Hence a lower bound on the total integral inventory over an inventory cycle is:

$$k \sum_{i=1}^{\lfloor M_w \rfloor} (\lfloor M_w \rfloor - i) R_w D R_w + \lfloor M_w \rfloor R_w D R_w = \frac{k(\lfloor M_w \rfloor - 1) \lfloor M_w \rfloor R_w^2 D}{2} + \lfloor M_w \rfloor R_w^2 D \tag{4.5}$$

Here, the second term on the left-hand-side accounts for the instance where the integral inventory begins at the higher level. The average integral inventory I_i per unit time satisfies:

$$I_i \geq \frac{k\left(\lfloor M_w \rfloor - 1\right)\lfloor M_w \rfloor}{2kM_w R_w} R_w^2 D + \frac{\lfloor M_w \rfloor R_w^2 D}{kM_w R_w} = \frac{(\lfloor M_w \rfloor - 1)\lfloor M_w \rfloor R_w D}{2M_w} + \frac{\lfloor M_w \rfloor R_w D}{kM_w} \tag{4.6}$$

Combining all the three parts of the inventory, we can see that the average inventory holding cost per unit time at the warehouse is: $(I_e + I_f + I_i)h_w$. It can be easily shown that the average inventory holding cost per unit time at the supplier over one production cycle is: $(1/2)pM_w R_w D^2 h_s$.

Now consider a new schedule where both the production cycle time at the supplier and the delivery cycle time from the supplier to the warehouse are increased to $\lceil M_w \rceil R_w$. In this schedule, there is no fractional or earliness inventory at the warehouse, as $\lceil M_w \rceil$ is an integer. Hence, the average inventory per unit time at the warehouse is $(1/2)\lfloor M_w \rfloor R_w D$. The average inventory per unit time at the supplier is $(1/2)p\lceil M_w \rceil R_w D^2 h_s$. Clearly, the average inventory per unit time at the customer, and the transportation cost from the warehouse to the customer in this new schedule remain the same as in the original schedule. Both transportation and production setup costs per unit time at the supplier are lower in this new schedule than in the original schedule as these activities are carried out less frequently. Therefore, the difference between the average total costs per unit time for the two schedules, denoted as Δ, satisfies:

$$\begin{aligned}
\Delta &\geq (I_e + I_f + I_i)h_w + \frac{1}{2}pM_w R_w D^2 h_s - \frac{1}{2}((\lceil M_w \rceil - 1)h_w + p\lceil M_w \rceil D h_s)R_w D \\
&\geq \left(\frac{(k-1)\lfloor M_w \rfloor}{2kM_w} + \frac{(k-1)}{2k} + \frac{(\lfloor M_w \rfloor - 1)\lfloor M_w \rfloor}{2M_w} + \frac{\lfloor M_w \rfloor}{kM_w} - \frac{\lfloor M_w \rfloor}{2} \right) R_w D h_w \\
&\quad + \frac{1}{2}pM_w R_w D^2 h_s - \frac{1}{2}p\lceil M_w \rceil R_w D^2 h_s \\
&= \frac{h_w R_w D}{2}\left(\frac{(1-a)\lfloor M_w \rfloor}{kM_w} + \frac{k-1}{k} \right) - \frac{pDh_s(k-a)R_w D}{2k} \qquad (4.7)
\end{aligned}$$

Since by model assumptions $h_s \leq h_w$ and $pD \leq 1$, we have $pDh_s \leq h_w$. Therefore (4.7) implies:

$$\begin{aligned}
\Delta &\geq \frac{h_w R_w D}{2}\left(\frac{(1-a)\lfloor M_w \rfloor}{kM_w} + 1 - \frac{1}{k} - \left(1 - \frac{a}{k}\right) \right) \\
&= \frac{h_w R_w D}{2}\frac{(a-1)}{k}\left(1 - \frac{\lfloor M_w \rfloor}{M_w}\right) \geq 0 \qquad (4.8)
\end{aligned}$$

This proves that the total cost of the new schedule is not greater than that of the original schedule. In the multiple-supplier case, we can generate a new schedule by increasing both T_i and R_i to $R_w\lceil R_i/R_w\rceil$ whenever (R_i/R_w) is not an integer. By the above proof, the total cost related to each product $i \in \{1,\ldots,m\}$ is not greater than that in the original schedule. Thus the average total cost in this new schedule is not greater than that in the original schedule. ■

By Theorem 16, we can focus on schedules where R_i/R_w is integer valued for each supplier $i = 1,\ldots,m$. In the following, we derive the average total cost per unit time in such a schedule. We have seen in the proof of Theorem 16 how to calculate the average inventory costs at the supplier and at the warehouse for the single-supplier case when R/R_w is integer valued. Extending that to the case with multiple suppliers, we get the following equation for the average total inventory cost per unit time, denoted as IC:

$$IC = \frac{1}{2}\left(\sum_{i=1}^{m} h_{si}p_i D_i^2 R_i\right) + \frac{1}{2}\left(\sum_{i=1}^{m} h_{wi}D_i(R_i - R_w)\right) + \frac{1}{2}\left(\sum_{i=1}^{m} h_{ci}D_i\right)R_w \quad (4.9)$$

where the first, second, and third term corresponds to the average inventory cost per unit time at the suppliers, at the warehouse, and at the customer, respectively.

The average total production setup cost per unit time, denoted as SC, and the average total distribution cost per unit time, denoted as DC, are given below:

$$SC = \sum_{i=1}^{m} \frac{S_i}{T_i} = \sum_{i=1}^{m} \frac{S_i}{R_i} \quad (4.10)$$

$$DC = \sum_{i=1}^{m} \frac{A_i}{R_i} + \frac{A_w}{R_w} \quad (4.11)$$

Therefore, the average total cost per unit time, denoted as TC, is given as:

$$\begin{aligned}
TC &= IC + SC + DC \\
&= \sum_{i=1}^{m} \frac{S_i}{R_i} + \frac{1}{2}\left(\sum_{i=1}^{m} h_{si} p_i D_i^2 R_i\right) + \frac{1}{2}\left(\sum_{i=1}^{m} h_{wi} D_i (R_i - R_w)\right) \\
&\qquad\qquad + \frac{1}{2}\left(\sum_{i=1}^{m} h_{ci} D_i\right) R_w + \sum_{i=1}^{m} \frac{A_i}{R_i} + \frac{A_w}{R_w} \\
&= \sum_{i=1}^{m} \frac{(S_i + A_i)}{R_i} + \sum_{i=1}^{m} \alpha_i R_i + \beta R_w + \frac{A_w}{R_w} \\
&= \sum_{i=1}^{m} \frac{Q_i}{R_i} + \sum_{i=1}^{m} \alpha_i R_i + \beta R_w + \frac{A_w}{R_w} \qquad (4.12)
\end{aligned}$$

where $Q_i = S_i + A_i$, $\alpha_i = (h_{si} p_i D_i + h_{wi}) D_i/2$, and $\beta = \sum_{i=1}^{m} (h_{ci} - h_{wi}) D_i/2$.

Our objective is to find the values for R_w and $R_1, \ldots, R_m$ that minimizes TC subject to the production constraint. The production constraint requires that the production cycle time T_i at each supplier i is sufficient to produce the required quantity along with the setup time, i.e. $s_i + p_i D_i T_i \leq T_i$. This means that $T_i \geq \tau_i$, i.e. $R_i \geq \tau_i$, as $R_i = T_i$ under policy (i), where $\tau_i = s_i/(1 - p_i D_i)$. So we can formulate our problem under policy (i) as follows:

$$\text{Minimize} \quad TC \qquad (4.13)$$

$$\text{Subject to:} \quad R_i \geq \tau_i, \ \forall i \in \{1, \ldots, m\} \qquad (4.14)$$

$$\frac{R_i}{R_w} \text{ is a positive integer}, \forall i \in \{1, \ldots, m\} \qquad (4.15)$$

$$R_w \geq 0 \qquad (4.16)$$

The average total cost per unit time TC given in equation (4.12) is a separable function of the variables R_w and $R_1, \ldots, R_m$, and corresponding to each variable, the function is convex. Hence without the constraint (4.15), the above formulation can be solved optimally using the first order Karush-Kuhn-Tucker (KKT) conditions. But the presence of the integrality constraint makes this problem more complicated. In the next two sections, we will show a way to obtain the optimal solution in the case of a single supplier and propose a heuristic for the multiple-supplier case.

4.2.2 Optimal Solution for the Single-Supplier Case

In this section, we derive the optimal solution in the case where there is only one supplier. For simplicity, we drop the supplier subscript from our notation. The total cost given in (4.12) can be rewritten as:

$$TC = \frac{Q}{R} + \alpha R + \beta R_w + \frac{A_w}{R_w} \tag{4.17}$$

where $Q = S + A$, $\alpha = (h_s pD + h_w)D/2$, and $\beta = (h_c - h_w)D/2$. Our objective is to find the values of R and R_w that minimize TC subject to the constraints that $R \geq \tau$ and R/R_w is a positive integer, where $\tau = s/(1 - pD)$.

Define $R' = \max\left\{\sqrt{Q/\alpha},\ \tau\right\}$ and $R'_w = \sqrt{A_w/\beta}$. If we ignore the integrality constraint, it can be shown by the first order KKT conditions that the optimal solution to this problem is given by $R = R'$ and $R_w = R'_w$. Let $M'_w = R'/R'_w$. Then we have the following result:

Theorem 17 Let R^* and R^*_w be the optimal values of R and R_w for the single-supplier problem under policy (i). Then,

$$R^*_w = \text{Max}\left\{\sqrt{\frac{Q + M^*_w A_w}{M^*_w(\alpha M^*_w + \beta)}},\ \frac{\tau}{M^*_w}\right\} \tag{4.18}$$

$$R^* = M^*_w R^*_w \tag{4.19}$$

where $M^*_w \in \{\max\{\lfloor M'_w \rfloor, 1\},\ \lceil M'_w \rceil\}$.

Proof We prove this by transforming our problem to an equivalent problem studied by Hahm and Yano (1992). Their supply chain consists of one supplier and one customer, with no warehouse in-between. We use the subscript y to denote the parameters involved in their problem. Their objective is to find the production cycle time T_y at the supplier and the delivery cycle time R_y from the supplier to the warehouse such that the average total cost per unit time is minimized. They show that in an optimal schedule, T_y is an integer multiple of R_y, and formulate their problem as the following optimization model:

$$
\begin{aligned}
\text{Minimize} \qquad & \frac{S_y}{T_y} + \frac{1}{2}(1 - p_y D_y) D_y h_y T_y + p_y D_y^2 h_y R_y + \frac{A_y}{R_y} \\
\text{Subject to:} \qquad & T_y \geq \frac{s_y}{1 - p_y D_y} \\
& \tfrac{T_y}{R_y} \text{ is a positive integer} \\
& R_y \geq 0
\end{aligned}
$$

where T_y and R_y are the decision variables and every other notation represents a problem parameter in the same way as the corresponding notations in our problem. In their problem, the unit holding costs at the supplier and at the customer are assumed to be equal and is represented by h_y.

This formulation is identical to our formulation with the following substitutions:

$$S_y = S + A \tag{4.20}$$

$$A_y = A_w \tag{4.21}$$

$$\frac{1}{2}(1 - p_y D_y) D_y h_y = \frac{1}{2}(h_s p D + h_w) D \tag{4.22}$$

$$p_y D_y^2 h_y = \frac{1}{2}(h_c - h_w) D \tag{4.23}$$

$$\frac{s_y}{1 - p_y D_y} = \frac{s}{1 - pD} \tag{4.24}$$

If we are able to find non-negative S_y, s_y, p_y, D_y, A_y, and h_y satisfying (4.20) - (4.24) and the capacity constraint $p_y D_y \leq 1$, then we can use the optimal solution from the Hahm and Yano model as the optimal solution for our model. The optimal solution for their problem is what we have given in equations (4.18) and (4.19) (with the corresponding substitutions of the parameters).

It can be easily shown that S_y, A_y, D_y, p_y, h_y, and s_y that are defined by (4.20), (4.21), and the following equations, respectively, satisfy (4.20) - (4.24) and $p_y D_y \leq 1$:

$$\begin{aligned} D_y &= D \\ p_y &= \frac{h_c - h_w}{2h_s p D^2 + h_c D + h_w D} \\ h_y &= h_s p D + \frac{1}{2}(h_c + h_w) \\ s_y &= \frac{2s}{(1-pD)} \frac{(h_s p D + h_w)}{(2h_s p D + h_c + h_w)} \end{aligned}$$

This completes the proof. ■

By Theorem 17, if $\text{Max}\{\lfloor M'_w \rfloor, 1\} = \lceil M'_w \rceil$, then the optimal solution is uniquely defined by (4.18) and (4.19). Otherwise, we only need to compare two solutions, one with $M^*_w = \text{Max}\{\lfloor M'_w \rfloor, 1\}$ and the other with $M^*_w = \lceil M'_w \rceil$, and the one with a lower objective value TC is the optimal solution to the problem.

4.2.3 A Heuristic Solution for the Multiple-Supplier Case

We first give an alternate representation for the problem defined in (4.13) - (4.16), where we substitute the variables R_i by $M_{wi} R_w$, for $i = 1, \ldots, m$:

$$\text{Minimize} \quad \sum_{i=1}^{m} \frac{Q_i}{M_{wi} R_w} + \sum_{i=1}^{m} \alpha_i M_{wi} R_w + \beta R_w + \frac{A_w}{R_w} \tag{4.25}$$

$$\text{Subject to:} \quad M_{wi} R_w \geq \tau_i, \ \forall i \in \{1, \ldots, m\} \tag{4.26}$$

$$M_{wi} \text{ is a positive integer, } \forall i \in \{1, \ldots, m\} \tag{4.27}$$

$$R_w \geq 0 \tag{4.28}$$

The heuristic in this section tries to find a near optimal solution $(M_{w1}, \ldots, M_{wm}, R_w)$ to the above formulation. Since there are multiple suppliers involved, the choice of M_{wi} at one supplier can influence the choice of M_{wi} at another supplier. If we just try two values of M_{wi} at each supplier as in the optimal solution for the single-supplier case shown in the previous subsection, the resulting solution is unlikely to be optimal or even local optimal. Our heuristic keeps trying different values of M_{wi}'s for the suppliers until a local optimal solution is found. More specifically, in each iteration, the heuristic chooses one supplier and fixes the M_{wi} values for all the other $m-1$ suppliers. Then it finds the values for R_w, R_i, and M_{wi} for the chosen supplier using an approach similar to the one used for the single-supplier problem. If the resulting total cost is lower, then the value of M_{wi} for the chosen supplier is updated and fixed in the next several iterations along with the M_{wi} values at $(m-2)$ other suppliers. The procedure stops when no improvement is found in one round of iterations across all the suppliers.

Heuristic H1

<u>Step 1</u>: Set $R_i^0 = \text{Max}\left\{\sqrt{\frac{Q_i}{\alpha_i}},\ \tau_i\right\}$, $R_w^0 = \sqrt{\frac{A_w}{\beta}}$, $M_{wi}^0 = R_i^0/R_w^0$, for $i = 1, \ldots, m$. Let $j = 1$ be the index of the supplier to be considered next. Set the iteration counter $c = 0$, and the non-improvement counter $n = 0$. Set the total cost $TC^0 = \infty$.

<u>Step 2</u>: Set $c = c + 1$. For supplier j, let $M'_{wj} = R_j^0/R'_w$ where

$$R'_w = \text{Max}\left\{\sqrt{\frac{\sum_{i=1,i\neq j}^{m} \frac{Q_i}{M_{wi}^0} + A_w}{\sum_{i=1,i\neq j}^{m} \alpha_i M_{wi}^0 + \beta}},\quad \max\left\{\frac{\tau_i}{M_{wi}^0},\quad \forall i \neq j\right\}\right\} \tag{4.29}$$

Set $\bar{M}_{wj} = \max\left\{\lfloor M'_{wj}\rfloor, 1\right\}$ and get the corresponding $\bar{R}_w$ as follows:

$$\bar{R}_w = \text{Max}\left\{\sqrt{\frac{\sum_{i=1,i\neq j}^{m} \frac{Q_i}{M_{wi}^0} + \frac{Q_j}{\bar{M}_{wj}} + A_w}{\sum_{i=1,i\neq j}^{m} \alpha_i M_{wi}^0 + \alpha_j \bar{M}_{wj} + \beta}},\quad \frac{\tau_j}{\bar{M}_{wj}},\quad \max\left\{\frac{\tau_i}{M_{wi}^0},\quad \forall i \neq j\right\}\right\} \tag{4.30}$$

Calculate the total cost TC of the solution $(M_{w1}, \ldots, M_{wm}, R_w)$ with $M_{wj} = \bar{M}_{wj}$, $M_{wi} =$

M^0_{wi} for $i \neq j$, and $R_w = \bar{R}_w$. Similarly, set $\bar{M}_{wj} = \lceil M'_{wj} \rceil$ and get the corresponding $\bar{R}_w$ and the total cost, TC, of the corresponding solution. Choose the solution with a lower total cost. Let the total cost of this solution be TC_j.

Step 3: If $c < m$, let M^0_{wj} be equal to the $\bar{M}_{wj}$ corresponding to the solution generated in Step 2. If $c \geq m$ and $TC_j < TC^0$, let $TC^0 = TC_j$ and M^0_{wj} be equal to the $\bar{M}_{wj}$ corresponding to the solution generated in Step 2, and reset $n = 0$. If $c \geq m$ and $TC_j \geq TC^0$, let $n = n + 1$.

Step 4: If $n = m$, there has been no improvement for any supplier in the last m iterations, and hence STOP. Otherwise, if $j = m$, set $j = 1$, else set $j = j + 1$. Go to Step 2.

In the first m iterations, the heuristic finds an integer solution for the variables M_{wj} at each supplier. After these iterations, the heuristic tries to improve the existing feasible solution, choosing one supplier at a time. By the first order KKT conditions, it can be shown that the values $R^0_1, \ldots, R^0_m$ and R^0_w defined in Step 1 of the heuristic are optimal to the problem (4.13) - (4.16) without the integrality requirement (4.15). Hence the values $M^0_{w1}, \ldots, M^0_{wm}$ and R^0_w defined in Step 1 are optimal to the problem (4.25) - (4.28) if the integrality constraint (4.27) is relaxed. Similarly, under the condition that M_{wi} is fixed as M^0_{wi} for all $i \neq j$, it can be shown that the solution (M'_{wj}, R'_w) defined in Step 2 is optimal to the remaining problem (4.25) - (4.28) with the integrality constraint relaxed. Since M'_{wj} may not be an integer, in Step 2, two integer solutions of M_{wj} rounded from M'_{wj} are evaluated. It can be easily verified that if M^0_{wi} is an integer for all $i \neq j$, then the solution generated in Step 2 is feasible to the problem (4.25) - (4.28).

Before evaluating the performance of the heuristic H1 computationally, we derive a lower and an upper bound on the M_{wi}'s in an optimal solution to the problem (4.25) - (4.28).

For $i = 1, \ldots, m$, define $Y_i = \alpha_i R_i^0 + Q_i/R_i^0$, where R_i^0 is defined in Step 1 of the heuristic. Clearly, $R_i = R_i^0$ is the optimal solution and Y_i is the optimal objective value of the problem $\min\{\alpha_i Ri + Q_i/R_i \mid R_i \geq \tau_i\}$. Let Z^{H1} denote the objective value of the solution obtained by heuristic H1 for the problem (4.25) - (4.28). Then in any optimal solution of the problem (4.25) - (4.28), we have

$$\beta R_w + A_w/R_w \leq Z^{H1} - \sum_{i=1}^{m} Y_i$$

Since the left-hand side of the above inequality is a convex function of R_w, it implies that in any optimal solution of the problem (4.25) - (4.28), $R_w \in [R_w^L, R_w^U]$, where

$$R_w^L = \frac{\left(Z^{H1} - \sum_{i=1}^m Y_i\right) - \sqrt{\left(Z^{H1} - \sum_{i=1}^m Y_i\right)^2 - 4\beta A_w}}{2\beta}$$

$$R_w^U = \frac{\left(Z^{H1} - \sum_{i=1}^m Y_i\right) + \sqrt{\left(Z^{H1} - \sum_{i=1}^m Y_i\right)^2 - 4\beta A_w}}{2\beta}$$

Define $Y_w = \beta R_w^0 + A_w/R_w^0$, where R_w^0 is defined in Step 1 of the heuristic. Clearly Y_w is the optimal objective value of the problem $\min\{\beta R_w + A_w/R_w\}$. In any optimal solution of the problem (4.13) - (4.15), we have

$$\alpha_i R_i + Q_i/R_i \leq Z^{H1} - \sum_{j=1}^{m} Y_j + Y_i - Y_w, \quad \text{for } i = 1, \ldots, m$$

Since the left-hand side of the above inequality is a convex function of R_i, it implies that in any optimal solution of the problem (4.13) - (4.15), $R_i \in [R_i^L, R_i^U]$, for $i = 1, \ldots, m$, where

$$R_i^L = \max\left\{\frac{\left(Z^{H1} - \sum_{j=1}^m Y_j + Y_i - Y_w\right) - \sqrt{\left(Z^{H1} - \sum_{j=1}^m Y_j + Y_i - Y_w\right)^2 - 4\alpha_i Q_i}}{2\alpha_i}, \quad \tau_i\right\}$$

$$R_i^U = \max\left\{\frac{\left(Z^{H1} - \sum_{j=1}^m Y_j + Y_i - Y_w\right) + \sqrt{\left(Z^{H1} - \sum_{j=1}^m Y_j + Y_i - Y_w\right)^2 - 4\alpha_i Q_i}}{2\alpha_i}, \quad \tau_i\right\}$$

Based on the above-derived lower and upper bounds of R_w and $R_1, \ldots, R_m$, we can conclude that in any optimal solution to the problem (4.25) - (4.28), the value of M_{wi} is within the interval $[M_{wi}^L, M_{wi}^U]$, where $M_{wi}^L = \lceil R_i^L/R_w^U \rceil$, and $M_{wi}^U = \lfloor R_i^U/R_w^L \rfloor$.

Now we are ready to test the performance of the heuristic H1. We compare the solution generated by the heuristic with the optimal solution obtained through an enumerative approach which enumerates all possible positive integer values of M_{wi} within its lower and upper bounds $[M_{wi}^L, M_{wi}^U]$, for $i = 1, \dots, m$, and for each possible combination of the values of $(M_{w1}, \dots, M_{wm})$, solves the rest of the problem with one variable R_w. We note that given the values of M_{wi} for $i = 1, \dots, m$, the optimal value of R_w is given by

$$R_w = \text{Max}\left\{\sqrt{\frac{\sum_{i=1}^{m}\frac{Q_i}{M_{wi}} + A_w}{\sum_{i=1}^{m}\alpha_i M_{wi} + \beta}}, \quad \max\left\{\frac{\tau_i}{M_{wi}}, \quad \forall i \in \{1, \dots, m\}\right\}\right\} \tag{4.31}$$

Since the enumerative approach for generating the optimal solution is computationally very demanding, we only test problems with two and four suppliers. In all the test problems, the following parameters are kept constant: demand rate $D_i = 10$ units per time unit, setup time $s_i = 1$, and setup cost $S_i = 100$, for $i = 1, \dots, m$. We test two production rates at the suppliers as follows: (i) one with lower production rates where each p_i is uniformly generated from the interval $[0.02, 0.08]$; and (ii) the other with higher production rates where each p_i is uniformly generated from the interval $[0.002, 0.008]$. So, the higher production rate considered is on average 10 times that of the lower production rate. The other parameters involved are generated randomly as follows for $i = 1, \dots, m$.

- Unit holding cost of product i at supplier i, $h_{si} = h_s \in \{0.1, 1, 10\}$ $\forall i$. Unit holding cost of product i at the warehouse $h_{wi} = h_w = \gamma_1 h_s, \forall i$, where $\gamma_1 \in \{1, 5, 10\}$. Unit holding cost of product i at the customer $h_{ci} = h_c = \gamma_2 h_w, \forall i$, where $\gamma_2 \in \{1.1, 5, 10\}$.

- The transportation cost per delivery $A_i = F_i + V_i$ and $A_w = F_w + V_w$ where F_i and F_w represent fixed costs and V_i and V_w variable costs determined by the corresponding travel distances. The fixed cost components $F_i = F_w = \rho F_0, \forall i$ and w, where the transportation cost index $F_0 \in \{1, 10, 100\}$ and the fixed cost factor $\rho \in \{0.01, 0.1, 1\}$.

Thus a value of 0.01 for the fixed cost factor implies very low fixed cost, while a value of 1 implies relatively high fixed cost. The variable cost components V_i and V_w are generated as follows. We assume that the suppliers are symmetrically arranged along a vertical line above and below a horizontal line through the customer so that the distance between neighboring suppliers is 0.2 units, and the horizontal distance from each supplier to the customer is one unit. For example, if there are two suppliers, then the (x, y) coordinates of the two suppliers are $\{(0, -0.1), (0, 0.1)\}$ while those of the customer are $(1, 0)$. We consider three cases of the warehouse location: the warehouse is located at $(x, 0)$, where $x \in \{0.2, 0.5, 0.8\}$. The distance between any two locations is calculated as the Euclidean distance. The variable cost V_i and V_w are calculated as the product of the transportation cost index F_0 and the Euclidean distance between the respective origin and destination.

For a given problem instance, while all the suppliers have the same holding costs, setup costs, and fixed transportation costs, they differ in their processing times (and hence capacity) and the variable transportation costs. We calculate the relative percentage gap between the total cost of the heuristic solution and that of the optimal solution. The relative gap (%) is defined as $\frac{Z^{H1} - Z^*}{Z^*} \times 100\%$, where Z^{H1} is the objective of the solution provided by the heuristic H1 and Z^* is the optimal objective value. For each combination of parameters $(p_i, h_s, \gamma_1, \gamma_2, F_0, \rho, x)$, we test five different random instances. Table 4.1 shows the results of our computational tests. Due to space restrictions, we aggregate the results corresponding to the two different ranges of production rates (p_i), three different values of the holding costs at the supplier (h_s), and the three different fixed cost factors (ρ). We provide both the average and maximum gaps for each entry corresponding to a combination of $(\gamma_1, \gamma_2, F_0, x)$. Thus each entry corresponds to 90 different problem instances. The overall average gap has a very small value of 0.18%. The average gap with two suppliers is 0.13% while that with four suppliers is 0.23%. The maximum among all the random test instances is 7.83%. The heuristic is very fast. In most cases, the convergence was achieved within one round of iterations between all the suppliers. In none of the cases, the CPU time was more than one second. Therefore, we can conclude that the heuristic is capable of generating near optimal solutions quickly for almost all the problems tested.

In general, the performance of the heuristic improves as we increase the value of γ_2, the holding cost multiplier for the customer. This is intuitive since when the holding cost at the customer increases, the frequency of deliveries from the warehouse to the customer relative to the frequency of delivery from the suppliers to the warehouse goes up. In other words, the values of M_{wi}'s increase. Hence the difference in the solution value for a small deviation from the optimal value for M_{wi}'s would be small. Also, when we increase the value of x (i.e. move the warehouse closer to the customer), the gap in general decreases. The reason for this is the same as the earlier one. When the warehouse is closer to the customer, the values of M_{wi}'s increase and the heuristic performs better. In summary, the heuristic seems to perform very well for most of the cases tested.

4.3 The Model under Policy (ii)

In our analysis so far, we have had the restriction that there can be only one delivery from a supplier to the warehouse in each production cycle at the supplier. In this section, we study the model under policy (ii) which relaxes this constraint and allows for multiple deliveries to the warehouse per production cycle at a supplier. It should be noted that when multiple deliveries per production cycle are allowed at the suppliers, Theorem 16 does not necessarily hold. However, Policy (ii) requires a feasible schedule to satisfy that theorem, i.e. there are multiple deliveries from the warehouse to the customer per delivery from a supplier to the warehouse.

We first formulate the problem under policy (ii) as a mathematical program. The inventory calculations at the warehouse and the customer remain the same as before (see (4.9) and the explanations thereafter), i.e. the average inventory cost per unit time at the warehouse is $1/2(\sum_{i=1}^{m} h_{wi} D_i (R_i - R_w))$ and that at the customer is $1/2\left(\sum_{i=1}^{m} h_{ci} D_i\right) R_w$. The inventory calculations at the suppliers is more complicated. Figure 4.3 shows an example inventory cycle for a supplier i where $T_i = 3R_i$. When T_i is an integer multiple of R_i, the average inventory at the supplier can be shown to be $\frac{1}{2}(1 - p_i D_i) D_i T_i + \left(p_i D_i - \frac{1}{2}\right) D_i R_i$. For details on the derivation, refer to Hahm and Yano (1992).

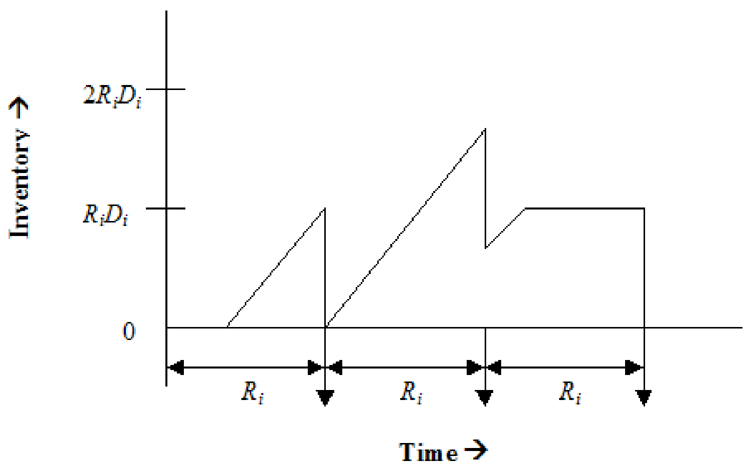

Figure 4.3: Inventory level at supplier i when $T_i = 3R_i$

The average total cost per time unit TC is thus given as follows:

$$TC = \sum_{i=1}^{m} \frac{S_i}{T_i} + \sum_{i=1}^{m} \alpha_i T_i + \sum_{i=1}^{m} \frac{A_i}{R_i} + \sum_{i=1}^{m} \beta_i R_i + \frac{A_w}{R_w} + \gamma R_w \tag{4.32}$$

where $\alpha_i = \frac{1}{2}(1 - p_i D_i) D_i h_{si}$, $\beta_i = \left(h_{si}\left(p_i D_i - \frac{1}{2}\right) + \frac{h_{wi}}{2}\right) D_i$, and $\gamma = \frac{1}{2}\sum_{i=1}^{m}(h_{ci} -$

$h_{wi})D_i$. Our problem under policy (ii) can be formulated as follows:

$$\text{Minimize} \quad TC \tag{4.33}$$

$$\text{Subject to:} \quad T_i \geq \tau_i, \ \forall i \in \{1, \dots, m\} \tag{4.34}$$

$$\frac{T_i}{R_i} \text{ is a positive integer, } \forall i \in \{1, \dots, m\} \tag{4.35}$$

$$\frac{R_i}{R_w} \text{ is a positive integer, } \forall i \in \{1, \dots, m\} \tag{4.36}$$

$$R_w \geq 0 \tag{4.37}$$

We propose a heuristic in Section 4.3.1 to solve this problem, and use this heuristic to study the value of warehouse in Section 4.3.2. Some other insights obtained from our computational experiments are provided in Section 4.3.3.

4.3.1 A Heuristic Solution

The idea of the heuristic is similar to that of heuristic H1 proposed in Section 2. The formulation (4.33) - (4.37) can be written as follows with $M_{s1}, \dots, M_{sm}$, $M_{w1}, \dots, M_{wm}$, and R_w as the decision variables:

$$\text{Minimize} \quad \sum_{i=1}^{m} \frac{S_i}{M_{si} M_{wi} R_w} + \sum_{i=1}^{m} \alpha_i M_{si} M_{wi} R_w + \sum_{i=1}^{m} \frac{A_i}{M_{wi} R_w}$$
$$+ \sum_{i=1}^{m} \beta_i M_{wi} R_w + \frac{A_w}{R_w} + \gamma R_w \tag{4.38}$$

$$\text{Subject to:} \quad M_{si} M_{wi} R_w \geq \tau_i, \ \forall i \in \{1, \dots, m\} \tag{4.39}$$

$$M_{si} \text{ and } M_{wi} \text{ are positive integers, } \forall i \in \{1, \dots, m\} \tag{4.40}$$

$$R_w \geq 0 \tag{4.41}$$

The heuristic tries to find a near optimal solution $(M_{s1}, \dots, M_{sm}, M_{w1}, \dots, M_{wm}, R_w)$ to the formulation (4.38) - (4.41) through the following iterative procedure: In each iteration, the heuristic chooses one supplier and fixes the M_{si} and M_{wi} values for all the other $m - 1$

suppliers. Then it solves the remaining problem with three decision variables R_w, M_{si}, and M_{wi} for the chosen supplier using an approach similar to the one in heuristic H1. In solving this problem, we try out four different combinations of M_{si} and M_{wi} values for the chosen supplier, and the best solution is used. If the resulting total cost is lower, then the value of M_{si} and M_{wi} for the chosen supplier are updated and fixed in the next several iterations. The procedure stops when no improvement is found in one round of iterations across all the suppliers.

Heuristic H2

Step 1: Set $T_i^0 = \text{Max}\left\{\sqrt{S_i/\alpha_i},\ \tau_i\right\}$, $R_i^0 = \sqrt{A_i/\beta_i}$, and $R_w^0 = \sqrt{A_w/\gamma}$, $M_{si}^0 = \frac{T_i^0}{R_i^0}$, $M_{wi}^0 = \frac{R_i^0}{R_w^0}$, for $i = 1, \ldots, m$. Set supplier index $j = 1$, the iteration counter $c = 0$, and the non-improvement counter $n = 0$. Set the total cost $TC^0 = \infty$.

Step 2: Set $c = c + 1$. For supplier j, let $\bar{M}_{sj} = \max\left\{\lfloor M_{sj}^0 \rfloor,\ 1\right\}$. Let $M'_{wj} = \frac{R'_j}{R'_w}$, where

$$R'_j = \text{Max}\left\{\sqrt{\frac{\frac{S_j}{\bar{M}_{sj}} + A_j}{\alpha_j \bar{M}_{sj} + \beta j}},\ \frac{\tau_j}{\bar{M}_{sj}}\right\} \tag{4.42}$$

$$R'_w = \text{Max}\left\{\sqrt{\frac{\sum_{i=1,i\neq j}^{m}\left(\frac{S_i}{M_{si}^0} + A_i\right)\frac{1}{M_{wi}^0} + A_w}{\sum_{i=1,i\neq j}^{m}(\alpha_i M_{si}^0 + \beta_i)M_{wi}^0 + \gamma}},\ \max\left\{\frac{\tau_i}{M_{si}^0 M_{wi}^0},\ \forall i \neq j\right\}\right\} \tag{4.43}$$

Set $\bar{M}_{wj} = \max\left\{\lfloor M_{wj}^0 \rfloor,\ 1\right\}$, and define $\bar{R}_w$ as follows:

$$\bar{R}_w = \text{Max}\left\{\sqrt{\frac{\sum_{i=1,i\neq j}^{n}\left(\frac{S_i}{M_{si}^0} + A_i\right)\frac{1}{M_{wi}^0} + A_w + \left(\frac{S_j}{\bar{M}_{sj}} + A_j\right)\frac{1}{\bar{M}_{wj}}}{\sum_{i=1,i\neq j}^{n}(\alpha_i M_{si}^0 + \beta_i)M_{wi}^0 + \gamma + (\alpha_j \bar{M}_{sj} + \beta_j)\bar{M}_{wj}}},\ \frac{\tau_j}{\bar{M}_{sj}\bar{M}_{wj}},\ \max\left\{\frac{\tau_i}{M_{si}^0 M_{wi}^0}\ \forall i \neq j\right\}\right\} \tag{4.44}$$

Calculate the total cost TC of the solution $(M_{s1}, \ldots, M_{sm}, M_{w1}, \ldots, M_{wm}, R_w)$ with $M_{sj} = \bar{M}_{sj}$, $M_{wj} = \bar{M}_{wj}$ at supplier j, $M_{si} = M_{si}^0$ and $M_{wi} = M_{wi}^0$ for $i \neq j$, and $R_w = \bar{R}_w$. Similarly, calculate the total costs of the other three solutions where only the values of

$(\bar{M}_{sj}, \bar{M}_{wj})$ are defined differently to be $\left(\max\left\{\lfloor M_{sj}^0 \rfloor, 1\right\}, \lceil M_{wj}^0 \rceil\right)$, $\left(\lceil M_{sj}^0 \rceil, \max\left\{\lfloor M_{wj}^0 \rfloor, 1\right\}\right)$, and $\left(\lceil M_{sj}^0 \rceil, \lceil M_{wj}^0 \rceil\right)$, respectively. Choose the solution with the lowest total cost. Let the total cost of this solution be TC_j.

Step 3: If $c \leq m$, let M_{sj}^0 and M_{wj}^0 be equal to the $\bar{M}_{sj}$ and $\bar{M}_{wj}$ corresponding to the solution chosen in Step 2. If $c \geq m$ and $TC_j < TC^0$, let $TC^0 = TC_j$ and let M_{sj}^0 and M_{wj}^0 be equal to the $\bar{M}_{sj}$ and $\bar{M}_{wj}$ corresponding to the solution chosen in Step 2, and reset $n = 0$. If $c \geq m$ and $TC_j \geq TC^0$, let $n = n + 1$.

Step 4: If $n = m$, there has been no cost improvement for any supplier in the last set of iterations, and hence STOP. Otherwise, if $j = m$, set $j = 1$, else set $j = j + 1$. Go to Step 2.

Similar to heuristic H1, in the first m iterations, we find an integer solution for the two variables (M_{sj}, M_{wj}) at each supplier. After these iterations, we try to improve the existing feasible solution, choosing one supplier at a time.

We evaluate the performance of heuristic H2 by comparing the solution generated by it with the optimal solution obtained by an enumerative approach which enumerates all possible positive integer values of M_{si} and M_{wi} within their valid lower and upper bounds, for $i = 1, \ldots, m$. We will discuss how to generate a valid lower and upper bound for each of these variables in the paragraphs that follow. Given the values of these variables, the optimal value of R_w is given by

$$R_w = \text{Max}\left\{\sqrt{\frac{\sum_{i=1}^n \left(\frac{S_i}{M_{si}} + A_i\right)\frac{1}{M_{wi}} + A_w}{\sum_{i=1}^n (\alpha_i M_{si}^0 + \beta_i) M_{wi} + \gamma}}, \quad \text{Max}\left\{\frac{\tau_i}{M_{si} M_{wi}}, \quad i = 1, \ldots, m\right\}\right\} \quad (4.45)$$

A lower and upper bound of each M variable in the formulation (4.38) - (4.41) can be derived in the same way as what we have done for the M variables in the formulation (4.25)

- (4.27) in Section 2. For $i = 1, \ldots, m$, define $U_i = \alpha_i T_i^0 + S_i/T_i^0$, and $V_i = \beta_i R_i^0 + A_i/R_i^0$, and define $W = \gamma R_w^0 + A_w/R_w^0$, where T_i^0, R_i^0, and R_w^0 are defined in Step 1 of heuristic H2. Let Z^{H2} denote the objective value of the solution obtained by heuristic H2 for the problem (4.38) - (4.41). Then in any optimal solution of the problem, we have

$$\begin{aligned}
\gamma R_w + A_w/R_w &\leq Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) \\
\beta_j R_j + A_j/R_j &\leq Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + V_j - W, \quad \text{for } j = 1, \ldots, m \\
\alpha_j T_j + S_j/T_j &\leq Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + U_j - W, \quad \text{for } j = 1, \ldots, m
\end{aligned}$$

which imply that in any optimal solution of the problem (4.38) - (4.41), $R_w \in [R_w^L, R_w^U]$, $R_j \in [R_j^L, R_j^U]$ and $T_j \in [T_j^L, T_j^U]$, for $j = 1, \ldots, m$, where

$$R_w^L = \frac{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i)\right) - \sqrt{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i)\right)^2 - 4\gamma A_w}}{2\gamma}$$

$$R_w^U = \frac{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i)\right) + \sqrt{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i)\right)^2 - 4\gamma A_w}}{2\gamma}$$

$$R_j^L = \frac{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + V_j - W\right) - \sqrt{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + V_j - W\right)^2 - 4\beta_j A_j}}{2\beta_j}$$

$$R_j^U = \frac{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + V_j - W\right) + \sqrt{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + V_j - W\right)^2 - 4\beta_j A_j}}{2\beta_j}$$

$$T_j^L = \max\left\{\frac{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + U_j - W\right) - \sqrt{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + U_j - W\right)^2 - 4\alpha_j S_j}}{2\alpha_j},\ \tau_j\right\}$$

$$T_j^U = \max\left\{\frac{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + U_j - W\right) + \sqrt{\left(Z^{H2} - \sum_{i=1}^{m}(U_i + V_i) + U_j - W\right)^2 - 4\alpha_j S_j}}{2\alpha_j},\ \tau_j\right\}$$

Based on the above-derived lower and upper bounds of R_w, $R_1, \ldots, R_m$, and $T_1, \ldots, T_m$, we can conclude that in any optimal solution to the problem (4.38) - (4.41), the values of M_{si} and M_{wi}, for every $i = 1, \ldots, m$, are within the interval $[M_{si}^L, M_{si}^U]$ and $[M_{wi}^L, M_{wi}^U]$, respectively, where $M_{si}^L = \lceil T_i^L/R_i^U \rceil$, $M_{si}^U = \lfloor T_i^U/R_i^L \rfloor$, $M_{wi}^L = \lceil R_i^L/R_w^U \rceil$, and $M_{wi}^U = \lfloor R_i^U/R_w^L \rfloor$.

Now we are ready to test the performance of heuristic H2. The parameter settings for the experiments are exactly the same as the ones for heuristic H1. Since the computational complexity for the enumeration procedure for policy (ii) is much higher than the enumeration procedure under policy (i), it is not possible to test for four suppliers within reasonable amounts of computational time. Hence we test cases with two and three suppliers. The results are shown in Table 4.2. Here again, the results are very good. The overall average gap is 0.16% while the maximum gap is 4.61%. The average and maximum for the two supplier cases are 0.19% and 4.61% respectively, while that for the three supplier cases are 0.12% and 3.88% respectively. The heuristic is very fast and none of the test instances took more than one second of CPU time. We see a trend similar to that of H1. When the multiplier values (M_{si}'s and M_{wi}'s) are large (such as low holding cost at the warehouse with high holding cost at the customer), the gap is very close to zero.

4.3.2 The Value of Warehouse

In the supply chains we consider, there is a warehouse between the suppliers and the customer and the products from the suppliers are consolidated at the warehouse for delivery to the customer. It is well-understood conceptually that the presence of a warehouse can lower the transportation and inventory costs compared to a single-stage supply chain where there is no warehouse between the suppliers and the customer. There are several simulation studies that compare freight consolidation through a warehouse and direct shipments based on transportation costs. For example, the study by Bagchi and Davis (1988) shows that direct shipments from vendors are almost always more expensive. Cooper (1984) compares freight consolidation across time and customers, use of warehouses, and direct less-than-truckload distribution systems on the basis of distribution costs and delivery times for selected product characteristics and demand patterns. She concludes that in general consolidation lowers costs but this may lead to an increase in the delivery time. These existing studies focus on transportation costs only and do not consider production operations and costs in the system. To our knowledge, no existing studies have investigated the value of consolidation or warehouses from a total system cost point of view. In this section, we computationally evaluate the typical reduction of total production, inventory and transportation cost that can be achieved by the use of a warehouse in the supply chain we consider. More specifically, we compare the total costs per unit time for the following two supply chains:

1) The two-stage supply chain considered in this chapter where there are m suppliers, one warehouse, and one customer;

2) A single-stage supply chain where there are also m suppliers and one customer as in our supply chain, but with no warehouse between the suppliers and the customer, and the product at each supplier is directly delivered to the customer.

We define the relative cost reduction with the addition of a warehouse as $\frac{Z_1^*-Z_2^*}{Z_1^*} \times 100\%$, where Z_1^* and Z_2^* are the optimal total cost per unit time in the single-stage and two-stage supply chains, respectively. Since there is no delivery consolidation in the single-stage supply chain, that problem can be viewed as m separate single-supplier problems, each equivalent to the model considered by Hahm and Yano (1992). Therefore, we solve the m separate single-stage single-supplier problems optimally by applying the solution approach of Hahm and Yano, and get the optimal total cost Z_1^*. We use heuristic H2 to solve the problem with the two-stage supply chain, and use the total cost of the solution obtained by H2, denoted as Z^{H2}, to replace Z_2^* in calculating the relative cost reduction. Thus the relative cost reduction we get in our computational test is a lower bound of the actual relative cost reduction.

In our computational experiment, we test three sets of problems with 2, 4, and 8 suppliers, respectively. All other parameters are generated exactly the same way as in the earlier experiments used for testing the performance of heuristics H1 and H2. For each set of the test problems, there are $2 \times 3^6 = 1458$ possible combinations of the parameters $(p_i, h_s, \gamma_1, \gamma_2, F_0, \rho, x)$, and for each combination of these parameters, we run ten random problem instances. For every test problem, the relative gap (%) between Z_1^* and Z^{H2} is computed. Table 4.3 through Table 4.5 show the results for the two cases of the supplier production rates for the 2-, 4-, and 8-supplier cases, respectively. The tables aggregate the results for the three different values of holding costs at the supplier (h_s), and hence each entry corresponds to an average over 30 random test instances for the given combination of the six parameters $(p_i, \gamma_1, \gamma_2, F_0, \rho, x)$.

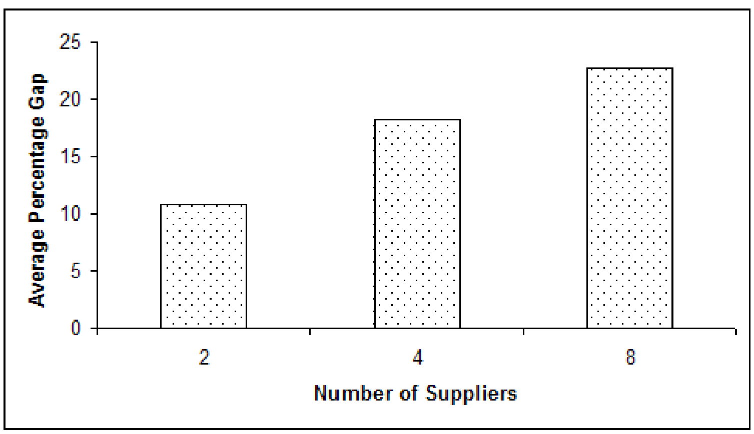

Figure 4.4: Average improvement with the warehouse

The warehouse serves as a place for pooling deliveries for commodities from the various suppliers in addition to acting as a place for holding inventory. Hence we would expect the relative gap to increase as we increase the number of suppliers. This is supported in the results obtained. As shown in Fig. 4.4, the overall average gap values for the two, four, and eight supplier cases are 10.75%, 18.18%, and 22.71% respectively. The few negative values in the tables indicate instances where adding a warehouse increases the total costs. Again we notice that the magnitude and the occurrence of negative values in general go down when we increase the number of suppliers. For all the results, in general, the relative gap increases as the holding cost at the customer goes up, or when the variable transportation cost goes up. The gap also increases with a reduction in the fixed cost factor value. The explanations for these are straightforward. All these changes make the warehouse an inexpensive transit point. The effect of the location of the warehouse on total cost is more involved. When both the holding cost at the customer and the variable transportation cost are high, we would like to have small and frequent shipments to save on the holding costs at the customer, but would like to avoid traveling long distances to contain the transportation expense. So it is advantageous to have the warehouse at close proximity to the customer site. On the other hand, if the holding cost at the customer is low when the transportation cost is high, we would like to make large shipments from the warehouse to the customer anyway, and hence placing the warehouse close to the supplier would reduce the delivery expenses from the supplier to the warehouse. Hence we see that in the tables, the relative gap increases with the value of x for the first case, and the gap decreases with x for the second. Finally, we can see from these tables that the production rates of the suppliers have some impact on the relative gap, but not as significant as the other parameters. Hence the suppliers' production rates do not seem to play a critical role in deciding whether to use a warehouse or not in the supply chain.

4.3.3 Other Insights

In this subsection, we discuss some other insights obtained from our computational experiments.

Theoretically, policy (ii) should always give solutions that are at least as good as the ones under policy (i) in terms of total cost. Since our heuristics do not guarantee to generate optimal solutions, we can not assert that H2 will always dominate H1. But in general, since both the heuristics have been shown to perform well in the computational experiments, we can expect H2 to provide better solutions than H1 in most cases. That is the reason why we chose H2 as the benchmark for the two-stage supply chain while evaluating the value of warehouse in the previous subsection. In fact, a comparison between the solutions obtained by H1 and H2 justify this choice. Due to the lack of space, we do not give any detailed reports on this. Instead, we provide a summary of our findings. Out of a total of 14580 test problems for the two supplier case, there were only 6 cases where H1 beat H2. Even in those cases, the gaps are very small in magnitude. The same holds true for the four supplier case. With the eight supplier problem instances, there was not even a single test problem for which H1 gave a total cost that was lower than the one given by H2. On average, the objective values from H2 were $28.36\%, 29.85\%$, and 30.24% lower than the objective values from H1 for the two, four, and eight supplier cases respectively.

Another interesting insight obtained is about the multipliers M_{si} in the model under policy (ii). A multiplier M_{si} represents the number of delivery cycles from supplier i to the warehouse per production cycle at supplier i. Table 4.6 shows the average number of M_{si}'s aggregated over the three cases of the number of suppliers (m) and the three cases of the holding cost at the suppliers (h_{si}).

We would expect the multipliers to depend on the location of the warehouse. For example, if the warehouse is closer to the suppliers, then the transportation cost from the suppliers to the warehouse is going to be lower and hence there will be more shipments from the suppliers per production cycle. This is supported in the results as the multipliers drop when the warehouse location is moved from $x = 0.2$ to $x = 0.5$ or $x = 0.8$. Similarly, we would expect the multiplier to decrease with the variable or fixed transportation cost. This is reflected in the table as the multipliers drop with ρ or F_0. We would expect the shipments from the suppliers to become more frequent (hence smaller) as the holding cost at the warehouse is increased. This observation is also supported in the table by the fact that the multipliers increase with γ_1.

Production rates can also play a crucial role. Production rates influence both the production batch size and the inventory holding costs. If the production is too slow along with a positive setup time, this may place restrictions on the batch size hence leading to higher costs. The effect of production rates on the inventory holding cost is slightly more complicated. A higher production rate may in fact be undesirable. This is because in the case of higher production rates, an item that is meant for a future shipment gets ready at an earlier time compared to a system with a lower production rate. This leads to increased waiting time for that item before getting shipped, thus resulting in an increased inventory holding cost at the supplier. Hence a higher production rate has both positive and negative impacts. As mentioned in Section 2, many studies in the past assumed infinite production rates at the suppliers. But Table 4.6 shows that changes in the production rate can significantly impact the number of delivery cycles from the suppliers per production cycle. When the production rates are increased by a factor of ten, the production-delivery cycle time ratio drops significantly. In many cases, the ratio drops by more than 75% of its original value.

4.4 Conclusions

In this study, we have studied a joint cyclic production and distribution scheduling problem in a two-stage supply chain with one or more suppliers, one warehouse, and one customer. We have given either optimal approaches or heuristic methods to solve the problem under two policies on production and delivery cycles. For the case with common production and delivery cycle at each supplier (policy (i)), we have proved that there exists an optimal solution where the delivery cycle time from a supplier to the warehouse is an integer multiple of the delivery cycle time from the warehouse to the customer. Based on this property, we have shown that there is a closed-form optimal solution to the problem with a single supplier under policy (i), and developed an efficient heuristic for the general problem under policy (i). The problem under policy (ii), which is more general than policy (i), is solved by a heuristic approach. Both heuristics are shown to perform very well for an extensive set of test problems. We have also computationally evaluated the value of warehouse in our two-stage supply chain. Various managerial insights have been reported.

An important use of this study is to make operational decisions regarding the delivery intervals in a two-stage supply chain. The approaches provided in this chapter are easy to implement. Moreover, computationally they are very efficient. The models can also be used to make strategic decisions related to configuring or making changes to a supply chain. For example, we could use the heuristics to choose between a single-stage and a two-stage supply chain. Given that a warehouse has to be built, we could use this study to analyze the total costs corresponding to various locations of the potential warehouse. We could use the heuristics to analyze the trade-offs involved in moving an existing warehouse to a new location. This model can also be used to analyze the effect of reducing the setup cost or setup time on the performance of the entire supply chain. For example, reducing the setup cost or setup time at the supplier would enable more frequent deliveries from the supplier to the warehouse, thus saving on the average inventory costs. A trade-off between this savings and the increase in the total transportation costs and the expenses related to reducing the setup time and costs can be used to analyze whether it is worth trying for a reduction in the setup cost or time.

Table 4.1: Average and maximum relative gaps (%) between the optimal solution and the solution provided by heuristic H1.

			$m = 2$						$m = 4$					
			Average Gap			Maximum Gap			Average Gap			Maximum Gap		
γ_1	γ_2	F_0	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$
		1	0.22	0.14	0.12	1.77	1.81	1.36	0.25	0.20	0.12	1.34	1.30	0.86
	1.1	10	0.45	0.41	0.14	4.87	4.09	1.80	0.58	0.47	0.43	3.42	3.33	3.54
		100	0.00	0.05	0.23	0.00	1.55	4.32	0.67	0.91	0.66	4.40	6.24	4.96
		1	0.01	0.02	0.02	0.16	0.23	0.26	0.04	0.03	0.03	0.23	0.17	0.17
1	5	10	0.05	0.04	0.03	0.50	0.82	0.27	0.07	0.05	0.04	0.42	0.33	0.24
		100	0.03	0.00	0.01	0.63	0.06	0.13	0.06	0.03	0.03	0.72	0.48	0.31
		1	0.01	0.01	0.01	0.10	0.12	0.13	0.02	0.02	0.01	0.13	0.10	0.11
	10	10	0.02	0.01	0.02	0.27	0.26	0.20	0.04	0.04	0.01	0.29	0.27	0.11
		100	0.02	0.01	0.01	0.60	0.13	0.22	0.04	0.02	0.02	0.47	0.23	0.23
		1	0.20	0.15	0.14	1.26	0.67	1.12	0.35	0.21	0.23	1.25	0.93	0.89
	1.1	10	0.32	0.36	0.24	2.29	2.00	2.06	0.60	0.50	0.45	2.64	2.87	1.93
		100	0.69	0.43	0.49	4.93	5.13	4.39	1.14	1.40	1.09	7.83	7.82	7.80
		1	0.03	0.02	0.02	0.18	0.14	0.15	0.03	0.03	0.03	0.13	0.17	0.15
5	5	10	0.06	0.05	0.04	0.52	0.35	0.27	0.08	0.05	0.05	0.61	0.30	0.30
		100	0.07	0.04	0.02	0.80	0.68	0.40	0.07	0.05	0.04	0.90	0.40	0.42
		1	0.02	0.02	0.01	0.11	0.09	0.08	0.02	0.03	0.02	0.11	0.13	0.07
	10	10	0.03	0.04	0.02	0.21	0.28	0.16	0.04	0.03	0.03	0.22	0.15	0.13
		100	0.03	0.03	0.03	0.33	0.25	0.28	0.05	0.04	0.03	0.44	0.49	0.38
		1	0.30	0.18	0.18	1.34	1.39	1.08	0.34	0.27	0.18	1.11	0.97	0.68
	1.1	10	0.70	0.51	0.29	3.75	2.42	2.09	0.72	0.62	0.55	2.79	2.64	2.29
		100	0.65	0.75	0.24	4.20	5.15	2.16	1.15	1.13	0.92	5.69	5.46	5.78
		1	0.02	0.03	0.02	0.12	0.17	0.20	0.05	0.04	0.04	0.15	0.14	0.12
10	5	10	0.05	0.06	0.04	0.36	0.56	0.36	0.10	0.07	0.06	0.37	0.23	0.33
		100	0.10	0.06	0.06	0.77	0.85	0.38	0.11	0.10	0.06	0.63	0.58	0.34
		1	0.02	0.02	0.02	0.11	0.09	0.10	0.02	0.02	0.02	0.09	0.07	0.08
	10	10	0.04	0.03	0.03	0.17	0.19	0.19	0.05	0.04	0.03	0.24	0.17	0.16
		100	0.05	0.04	0.03	0.44	0.34	0.28	0.07	0.06	0.03	0.52	0.41	0.15

Table 4.2: Average and maximum relative gaps (%) between the optimal solution and the solution provided by heuristic H2.

			$m = 2$						$m = 3$					
			Average Gap			Maximum Gap			Average Gap			Maximum Gap		
γ_1	γ_2	F_0	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$
		1	1.03	0.37	0.17	2.31	1.39	0.83	1.06	0.61	0.51	2.72	2.31	2.35
	1.1	10	1.45	0.16	0.04	3.49	0.74	0.47	0.86	0.19	0.18	2.70	1.03	3.06
		100	0.38	0.00	0.12	2.32	0.03	4.21	0.26	0.17	0.00	1.47	3.88	0.06
		1	0.41	0.11	0.02	1.32	0.34	0.40	0.19	0.06	0.02	0.79	0.35	0.24
1	5	10	0.30	0.02	0.02	1.47	0.58	0.16	0.06	0.01	0.00	0.51	0.17	0.02
		100	0.06	0.02	0.01	0.81	0.71	0.13	0.02	0.00	0.00	0.18	0.06	0.00
		1	0.14	0.02	0.01	0.64	0.26	0.16	0.08	0.03	0.01	0.37	0.14	0.05
	10	10	0.04	0.00	0.00	0.50	0.00	0.18	0.03	0.01	0.00	0.44	0.19	0.01
		100	0.01	0.00	0.00	0.34	0.03	0.01	0.00	0.00	0.00	0.11	0.06	0.00
		1	1.31	0.24	0.04	2.63	0.46	0.20	0.67	0.10	0.02	1.30	0.30	0.17
	1.1	10	1.64	0.20	0.02	4.47	0.46	0.22	0.73	0.07	0.00	1.72	0.26	0.02
		100	0.82	0.03	0.01	3.22	0.37	0.43	0.33	0.00	0.01	1.18	0.00	0.36
		1	0.28	0.02	0.02	1.01	0.06	0.14	0.13	0.09	0.01	0.41	0.26	0.05
5	5	10	0.22	0.00	0.01	1.27	0.05	0.13	0.04	0.05	0.00	0.21	0.16	0.03
		100	0.05	0.00	0.00	0.63	0.03	0.01	0.00	0.01	0.00	0.03	0.12	0.01
		1	0.17	0.01	0.01	0.57	0.03	0.02	0.13	0.03	0.01	0.37	0.09	0.10
	10	10	0.11	0.00	0.00	0.62	0.06	0.01	0.05	0.00	0.00	0.24	0.02	0.01
		100	0.02	0.00	0.00	0.15	0.05	0.01	0.00	0.00	0.00	0.05	0.05	0.01
		1	1.55	0.28	0.04	3.38	0.52	0.22	0.75	0.10	0.01	1.44	0.20	0.10
	1.1	10	1.65	0.23	0.01	4.61	0.58	0.15	0.78	0.08	0.00	1.74	0.21	0.04
		100	0.81	0.03	0.00	2.54	0.27	0.03	0.34	0.01	0.04	0.93	0.09	1.93
		1	0.30	0.01	0.02	1.08	0.05	0.13	0.12	0.07	0.00	0.40	0.14	0.04
10	5	10	0.21	0.00	0.01	1.14	0.05	0.21	0.05	0.03	0.00	0.25	0.11	0.03
		100	0.08	0.00	0.01	0.49	0.04	0.11	0.01	0.02	0.00	0.18	0.23	0.00
		1	0.18	0.02	0.01	0.57	0.06	0.03	0.13	0.02	0.01	0.40	0.06	0.15
	10	10	0.11	0.01	0.00	0.45	0.05	0.06	0.06	0.01	0.00	0.35	0.06	0.04
		100	0.03	0.00	0.00	0.30	0.00	0.04	0.00	0.00	0.00	0.09	0.00	0.00

Table 4.3: Relative cost reductions (%) due to the warehouse when there are two suppliers

				$\rho = 0.01$			$\rho = 0.1$			$\rho = 1$		
p_i	γ_1	γ_2	F_0	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$
			1	1.25	0.96	0.05	1.07	0.53	-0.25	-1.44	-1.75	-1.99
		1.1	10	4.23	3.48	1.19	3.42	2.66	0.47	-1.90	-3.02	-4.24
			100	10.97	6.73	2.25	8.73	4.91	0.71	-2.47	-4.57	-6.94
			1	4.60	5.65	7.74	4.19	4.99	6.77	1.72	1.94	2.79
	1	5	10	10.45	11.76	16.00	9.21	10.34	13.93	2.89	3.92	5.44
			100	16.55	17.53	24.31	14.21	15.59	20.95	4.10	6.04	6.92
			1	7.82	9.32	13.01	7.11	8.79	11.88	4.70	5.75	7.21
		10	10	14.71	17.78	24.52	13.48	16.41	22.35	7.99	10.23	12.69
			100	20.51	24.67	33.70	18.43	22.72	30.38	9.96	12.34	15.66
			1	3.49	2.97	1.23	2.99	2.17	0.56	-1.53	-2.25	-3.20
		5.5	10	8.41	6.27	2.44	6.92	4.61	1.16	-2.19	-3.83	-5.75
			100	14.65	9.65	3.63	11.88	7.19	1.70	-2.52	-5.13	-8.01
			1	8.62	9.67	12.92	7.80	8.56	11.53	2.33	3.24	4.18
U[0.02,0.08]	5	25	10	14.55	16.12	21.59	12.88	14.17	18.73	3.68	4.88	6.45
			100	18.71	20.29	27.29	16.29	17.75	23.35	4.38	5.79	7.91
			1	12.62	14.91	21.03	11.23	14.22	18.85	6.72	8.53	10.62
		50	10	18.92	22.75	31.37	16.98	20.82	28.06	9.48	11.88	14.81
			100	22.56	27.12	37.47	20.23	24.72	33.23	10.79	13.52	16.85
			1	4.68	3.85	1.55	3.88	2.81	0.74	-1.79	-2.82	-4.00
		11	10	10.29	7.49	2.89	8.44	5.48	1.34	-2.42	-4.45	-6.61
			100	15.90	10.36	3.92	12.78	7.67	1.85	-2.66	-5.40	-8.30
			1	10.41	11.76	15.81	9.42	10.42	13.81	2.77	3.79	4.90
	10	50	10	16.07	17.67	23.79	14.14	15.52	20.56	3.92	5.25	6.87
			100	19.40	21.06	28.29	16.89	18.41	24.25	4.55	6.04	7.89
			1	14.49	17.60	24.38	13.26	16.37	21.67	7.60	9.67	12.17
		100	10	20.24	24.50	33.80	18.27	22.43	29.97	9.98	12.54	15.65
			100	23.17	27.86	38.44	20.75	25.35	34.11	10.99	13.80	17.20
			1	0.72	0.47	0.60	0.68	0.30	0.45	-0.19	-0.34	-0.41
		1.1	10	2.83	2.06	2.12	2.40	1.70	1.68	-0.49	-1.06	-1.66
			100	9.82	6.15	2.35	8.21	4.74	1.16	-1.96	-4.16	-6.47
			1	5.19	6.77	9.46	5.14	6.61	8.89	4.92	6.01	7.08
	1	5	10	12.06	15.56	21.53	11.74	14.97	20.12	10.44	12.52	14.68
			100	19.87	24.59	32.71	18.64	22.70	29.43	11.62	13.28	15.76
			1	7.64	10.17	13.92	7.59	9.94	13.27	7.63	9.23	11.03
		10	10	15.84	21.13	29.14	15.69	20.52	27.32	14.54	17.45	20.37
			100	24.13	30.76	41.36	23.03	28.94	37.90	16.62	19.55	22.97
			1	2.59	2.18	0.85	2.17	1.63	0.42	-1.04	-1.57	-2.30
		5.5	10	6.86	5.05	1.94	5.72	3.86	0.92	-1.72	-3.13	-4.91
			100	12.94	8.27	3.14	11.09	6.27	1.53	-2.40	-5.33	-7.75
			1	7.43	8.34	11.34	6.74	7.57	10.04	2.64	3.52	4.57
U[0.002,0.008]	5	25	10	13.76	15.26	20.74	12.21	13.68	18.16	4.42	5.79	7.60
			100	18.72	20.72	28.00	16.29	18.35	24.40	5.62	7.40	9.29
			1	10.53	13.14	18.19	9.99	12.31	16.59	6.69	8.23	10.18
		50	10	17.64	21.79	30.11	16.39	20.18	27.18	10.11	12.45	15.38
			100	22.37	27.48	37.96	20.43	25.16	34.01	12.05	14.70	18.12
			1	3.57	2.96	1.17	3.00	2.20	0.57	-1.39	-2.15	-3.08
		11	10	8.50	6.32	2.40	6.97	4.69	1.16	-1.89	-3.75	-5.68
			100	14.63	9.68	3.54	11.88	7.02	1.71	-2.34	-4.96	-7.71
			1	8.95	10.05	13.60	8.12	9.04	11.93	2.61	3.75	4.86
	10	50	10	15.06	16.67	22.53	13.34	14.79	19.55	4.14	5.67	7.37
			100	19.23	21.02	28.36	16.77	18.53	24.47	4.96	6.64	8.77
			1	12.52	15.40	21.39	11.67	14.33	19.39	7.31	9.02	11.25
		100	10	19.08	23.29	32.25	17.49	21.42	28.92	10.17	12.54	15.61
			100	22.85	27.76	38.38	20.76	25.35	34.23	11.56	14.29	17.78

Table 4.4: Relative cost reductions (%) due to the warehouse when there are four suppliers

p_i	γ_1	γ_2	F_0	$\rho = 0.01$ $x = 0.2$	$\rho = 0.01$ $x = 0.5$	$\rho = 0.01$ $x = 0.8$	$\rho = 0.1$ $x = 0.2$	$\rho = 0.1$ $x = 0.5$	$\rho = 0.1$ $x = 0.8$	$\rho = 1$ $x = 0.2$	$\rho = 1$ $x = 0.5$	$\rho = 1$ $x = 0.8$
			1	2.07	1.07	-0.03	2.03	1.04	-0.16	0.29	-0.47	-1.44
		1.1	10	8.43	5.05	1.82	7.62	4.68	1.34	2.08	0.06	-1.83
			100	17.33	10.41	3.68	14.41	8.49	2.24	4.35	0.89	-2.52
			1	8.58	8.87	9.83	8.28	8.50	9.27	6.97	6.98	7.36
	1	5	10	18.49	18.26	20.34	17.48	17.45	19.07	13.23	13.31	13.66
			100	27.93	27.56	30.49	26.16	25.94	28.33	18.32	18.50	18.78
			1	12.72	13.81	15.94	12.66	13.53	15.33	11.81	12.50	13.22
		10	10	24.32	25.96	29.97	23.46	25.13	28.50	20.33	21.45	22.53
			100	33.79	35.69	41.39	32.30	34.32	39.07	26.39	27.25	29.32
			1	6.60	4.54	1.76	6.06	3.99	1.37	2.14	0.75	-0.90
		5.5	10	14.83	9.66	3.61	13.37	8.39	2.86	4.21	1.47	-1.62
			100	22.94	14.37	5.48	19.95	12.37	4.22	5.85	2.08	-2.32
			1	14.90	15.04	16.75	14.20	14.36	15.77	11.04	11.35	11.62
U[0.02,0.08]	5	25	10	25.17	24.91	27.60	23.59	23.56	25.46	16.72	17.07	17.58
			100	31.88	31.45	34.76	29.55	29.49	31.93	19.95	20.36	20.85
			1	20.71	22.03	25.51	20.07	21.55	24.18	17.69	18.67	19.63
		50	10	31.36	33.17	38.33	30.13	31.87	36.18	24.64	25.83	27.39
			100	37.40	39.64	45.67	35.54	37.82	42.73	28.01	29.46	31.30
			1	8.80	6.00	2.25	7.98	5.19	1.81	2.76	0.98	-1.10
		11	10	17.47	11.27	4.28	15.49	9.71	3.33	4.85	1.72	-1.78
			100	24.65	15.49	5.87	21.42	13.13	4.53	6.29	2.22	-2.32
			1	18.35	18.06	19.83	16.93	17.27	18.77	12.89	13.27	13.64
	10	50	10	27.65	27.36	30.18	25.49	25.89	28.11	17.99	18.37	18.90
			100	33.12	32.60	36.04	30.55	30.55	33.11	20.49	20.94	21.53
			1	24.29	25.56	29.56	23.50	24.75	28.29	20.19	21.13	22.38
		100	10	33.81	35.78	41.13	32.26	34.36	38.60	26.01	27.30	28.84
			100	38.41	40.71	46.91	36.48	38.80	43.83	28.58	30.05	31.82
			1	1.08	0.98	0.89	0.93	0.91	0.80	0.59	0.50	0.36
		1.1	10	4.27	3.02	2.84	3.85	2.93	2.65	1.52	1.10	0.56
			100	14.00	9.02	4.36	12.75	8.04	3.22	4.78	1.72	-1.74
			1	8.36	9.42	11.15	8.41	9.42	10.90	9.41	10.00	10.71
	1	5	10	19.25	21.59	25.53	19.19	21.26	24.81	19.96	21.17	22.66
			100	31.79	34.33	39.04	30.90	33.04	36.81	25.55	26.52	27.54
			1	12.08	13.76	16.32	12.22	13.76	16.03	13.71	14.70	15.82
		10	10	25.14	28.58	34.06	25.12	28.46	33.01	25.99	27.84	29.90
			100	37.67	41.89	48.60	36.90	40.63	46.23	32.34	34.08	36.07
			1	4.83	3.39	1.29	4.50	2.97	1.02	1.67	0.60	-0.61
		5.5	10	11.91	7.69	2.90	10.69	6.75	2.36	3.52	1.29	-1.44
			100	20.46	12.74	4.30	18.11	10.66	3.74	5.76	2.06	-2.61
			1	12.47	12.72	14.33	11.96	12.29	13.55	10.03	10.27	10.70
U[0.002,0.008]	5	25	10	23.06	23.30	26.16	21.87	22.21	24.45	16.62	17.09	17.79
			100	31.28	31.41	35.31	29.35	29.71	32.80	21.09	21.59	22.31
			1	17.72	19.08	22.14	17.37	18.64	21.19	16.04	16.94	18.02
		50	10	29.35	31.55	36.56	28.38	30.48	34.61	24.20	25.60	27.24
			100	36.99	39.73	46.02	35.49	38.00	43.29	28.84	30.42	32.25
			1	6.73	4.61	1.76	6.18	4.04	1.40	2.20	0.80	-0.82
		11	10	14.72	9.55	3.64	13.09	8.27	2.84	4.30	1.49	-1.52
			100	22.86	14.39	5.29	19.93	12.22	4.17	5.76	2.06	-2.07
			1	15.38	15.36	17.25	14.56	14.81	16.22	11.54	11.87	12.29
	10	50	10	25.68	25.57	28.50	24.05	24.23	26.49	17.51	17.94	18.55
			100	32.42	32.21	35.88	30.18	30.35	33.13	20.73	21.30	22.05
			1	21.10	22.56	26.10	20.48	21.91	24.89	18.24	19.22	20.42
		100	10	31.81	33.96	39.25	30.55	32.66	37.01	25.32	26.66	28.33
			100	37.87	40.40	46.68	36.15	38.60	43.77	28.84	30.32	32.23

Table 4.5: Relative cost reductions (%) due to the warehouse when there are eight suppliers

				$\rho = 0.01$			$\rho = 0.1$			$\rho = 1$		
p_i	γ_1	γ_2	F_0	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$
			1	1.03	0.82	0.61	1.06	0.88	0.24	0.65	-0.21	-0.78
		1.1	10	7.88	5.40	2.16	7.03	4.82	1.57	3.22	1.57	-0.51
			100	16.10	10.71	4.49	14.26	9.25	2.81	6.28	3.40	-0.12
			1	11.08	11.06	11.55	10.89	10.92	11.26	10.60	10.70	10.73
	1	5	10	22.85	22.83	23.66	22.36	22.25	22.74	20.09	20.13	20.20
			100	34.33	33.86	35.09	33.10	32.80	33.72	27.72	27.88	27.66
			1	16.35	17.06	18.31	16.44	17.06	18.09	16.95	17.32	17.74
		10	10	30.82	31.86	34.16	30.37	31.31	33.24	28.87	29.46	30.25
			100	42.31	43.77	47.03	41.37	42.64	45.09	37.13	37.96	38.65
			1	4.72	4.01	1.85	4.82	3.76	1.59	3.05	1.84	0.13
		5.5	10	13.12	9.34	4.15	12.14	8.46	3.48	6.04	3.58	0.37
			100	21.14	14.15	6.21	19.31	12.66	5.01	8.44	4.86	0.50
			1	19.01	18.87	19.35	18.43	18.53	18.76	16.92	16.87	16.86
U[0.02,0.08]	5	25	10	31.20	30.87	31.93	30.16	29.94	30.57	25.50	25.56	25.49
			100	39.06	38.58	39.99	37.43	37.16	38.04	30.41	30.40	30.32
			1	26.28	27.33	29.15	25.96	26.72	28.48	25.30	25.90	26.36
		50	10	39.25	40.49	43.43	38.61	39.61	42.00	34.99	35.74	36.52
			100	46.65	48.07	51.60	45.41	46.78	49.56	39.78	40.65	41.52
			1	6.81	5.50	2.45	6.61	5.06	2.09	3.91	2.33	0.21
		11	10	15.65	10.85	4.87	14.34	9.86	4.04	6.94	4.07	0.45
			100	22.73	15.07	6.68	20.53	13.52	5.49	8.95	5.21	0.59
			1	22.56	22.43	23.52	22.05	21.76	22.64	19.71	19.82	19.79
	10	50	10	34.05	33.70	34.88	32.78	32.69	33.39	27.37	27.43	27.39
			100	40.43	39.96	41.42	38.74	38.48	39.39	31.20	31.23	31.14
			1	30.64	31.69	34.04	30.20	31.07	33.22	28.72	29.32	29.98
		100	10	42.11	43.52	46.64	41.29	42.40	45.14	36.93	37.69	38.54
			100	47.83	49.35	52.94	46.53	47.99	50.81	40.57	41.44	42.32
			1	1.15	1.29	1.12	1.18	1.26	1.06	1.14	0.93	0.80
		1.1	10	4.48	3.97	3.48	4.19	3.84	3.40	2.94	2.42	1.86
			100	13.55	9.52	5.28	12.66	8.67	4.55	6.83	4.01	0.47
			1	10.79	11.54	12.65	10.96	11.65	12.63	12.70	13.13	13.58
	1	5	10	24.58	26.20	28.81	24.66	26.28	28.44	26.61	27.52	28.38
			100	39.50	41.04	43.78	38.88	40.19	42.27	35.17	35.58	35.99
			1	15.51	16.73	18.46	15.80	16.89	18.39	18.18	18.91	19.65
		10	10	32.01	34.48	38.07	32.24	34.50	37.61	34.28	35.58	36.91
			100	47.04	49.79	53.99	46.55	48.95	52.38	43.29	44.37	45.53
			1	3.55	3.16	1.55	3.74	2.94	1.34	2.41	1.44	0.17
		5.5	10	10.84	7.62	3.60	10.31	6.92	3.00	5.10	3.02	0.36
			100	19.20	12.52	5.89	17.44	10.96	5.00	8.21	4.81	0.57
			1	15.89	15.91	16.70	15.60	15.71	16.28	15.05	15.16	15.26
U[0.002,0.008]	5	25	10	28.78	28.81	30.19	28.00	28.12	29.11	24.93	25.09	25.21
			100	38.61	38.64	40.54	37.26	37.42	38.82	31.34	31.49	31.50
			1	22.61	23.47	25.31	22.48	23.31	24.83	22.72	23.26	23.85
		50	10	37.02	38.43	41.46	36.42	37.77	40.23	34.15	34.98	35.84
			100	46.32	48.10	51.92	45.26	46.94	50.05	40.54	41.47	42.39
			1	5.47	4.38	2.05	5.48	4.05	1.75	3.20	1.90	0.22
		11	10	13.62	9.41	4.28	12.56	8.51	3.60	6.15	3.55	0.42
			100	21.27	14.12	6.36	19.28	12.53	5.34	8.28	4.81	0.57
			1	19.38	19.29	20.11	18.90	18.94	19.51	17.57	17.64	17.66
	10	50	10	31.79	31.59	32.93	30.74	30.69	31.62	26.45	26.56	26.58
			100	39.81	39.56	41.23	38.28	38.18	39.33	31.42	31.51	31.59
			1	26.77	27.73	29.83	26.51	27.42	29.13	25.95	26.55	27.17
		100	10	39.97	41.35	44.49	39.16	40.49	43.02	35.83	36.66	37.50
			100	47.33	48.96	52.67	46.11	47.68	50.66	40.70	41.61	42.61

Table 4.6: Average number of deliveries from the suppliers to the warehouse per production cycle at the suppliers

				$\rho = 0.01$			$\rho = 0.1$			$\rho = 1$		
p_i	γ_1	γ_2	F_0	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$	$x = 0.2$	$x = 0.5$	$x = 0.8$
			1	28.88	19.98	16.39	24.04	18.68	15.67	12.81	11.90	11.22
		1.1	10	8.23	6.47	5.19	7.26	5.50	4.77	4.09	3.81	3.57
			100	2.53	1.89	1.67	2.22	1.82	1.57	1.21	1.16	1.14
			1	28.47	20.37	16.21	24.06	18.84	15.79	12.93	11.86	11.45
	1	5	10	8.52	6.32	5.21	7.38	6.11	4.96	4.10	3.79	3.62
			100	2.84	2.00	1.69	2.34	1.89	1.64	1.44	1.35	1.25
			1	28.42	20.12	16.97	23.67	18.81	15.75	13.50	12.02	10.64
		10	10	9.23	6.35	5.14	7.56	5.72	4.90	4.30	3.83	3.70
			100	2.76	2.05	1.73	2.53	1.97	1.55	1.39	1.34	1.27
			1	61.12	43.84	36.15	51.57	41.48	34.43	29.04	26.17	24.67
		5.5	10	19.33	13.63	11.48	16.41	12.68	11.21	9.02	8.09	7.56
			100	5.82	4.22	3.74	4.98	3.95	3.54	2.72	2.50	2.39
			1	62.48	43.52	36.65	53.32	41.92	34.25	29.34	26.80	24.97
U[0.02,0.08]	5	25	10	19.65	14.27	11.25	16.47	13.06	10.80	9.52	8.54	7.95
			100	5.97	4.37	3.70	5.19	4.14	3.43	3.11	2.62	2.50
			1	62.39	43.68	36.30	54.38	40.42	34.06	30.43	26.79	24.67
		50	10	19.37	14.14	11.47	16.83	13.17	10.97	9.36	8.76	7.93
			100	6.32	4.47	3.73	5.32	4.19	3.51	3.02	2.68	2.53
			1	85.99	62.00	52.11	74.76	57.64	49.01	40.00	37.87	34.74
		11	10	27.13	19.84	16.07	23.72	18.04	15.34	12.58	11.55	10.74
			100	8.34	6.14	4.98	7.17	5.52	4.92	3.87	3.62	3.43
			1	85.64	62.70	51.96	75.98	59.41	48.11	41.98	37.08	34.87
	10	50	10	27.27	19.93	15.99	23.45	18.40	15.33	13.22	11.76	11.20
			100	8.63	6.33	5.10	7.44	5.91	4.81	4.39	3.70	3.55
			1	85.88	63.37	51.94	73.93	59.35	50.08	41.30	38.11	35.52
		100	10	27.79	19.74	16.25	23.48	18.24	15.43	13.16	12.09	11.29
			100	8.86	6.43	5.23	7.54	5.88	5.00	4.09	3.79	3.65
			1	5.91	4.18	3.48	5.04	3.87	3.37	2.74	2.49	2.34
		1.1	10	1.74	1.31	1.01	1.53	1.18	1.00	1.00	1.00	1.00
			100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
			1	5.87	4.28	3.47	5.09	3.94	3.32	2.70	2.56	2.51
	1	5	10	1.85	1.44	1.05	1.63	1.35	1.00	1.00	1.00	1.00
			100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
			1	5.89	4.24	3.51	5.04	3.99	3.32	2.81	2.58	2.41
		10	10	1.96	1.42	1.03	1.64	1.27	1.00	1.00	1.00	1.00
			100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
			1	37.76	27.33	22.29	32.83	25.35	21.29	17.81	16.20	15.01
		5.5	10	11.69	8.50	7.18	10.25	7.92	6.57	5.18	5.00	4.54
			100	3.58	2.62	2.33	3.00	2.33	2.00	1.97	1.72	1.43
			1	37.82	27.67	22.58	32.86	25.74	21.32	17.90	16.59	15.30
U[0.002,0.008]	5	25	10	11.76	8.74	6.95	10.17	8.37	6.88	5.49	5.32	5.00
			100	3.83	2.86	2.00	3.11	2.78	2.00	2.00	2.00	1.98
			1	37.63	27.33	22.57	32.89	25.33	21.39	17.90	16.23	15.27
		50	10	11.62	8.76	6.93	9.99	7.86	6.86	5.70	5.00	4.99
			100	3.54	2.87	2.00	3.33	2.80	2.00	2.00	2.00	2.00
			1	56.40	40.74	33.35	49.04	37.84	31.73	26.63	24.41	22.51
		11	10	17.78	12.65	10.59	15.41	11.71	9.98	8.01	7.48	7.00
			100	5.43	4.00	3.33	4.58	3.79	3.00	2.16	2.00	2.00
			1	56.38	40.82	33.29	49.08	38.01	31.98	26.90	24.80	22.73
	10	50	10	17.78	12.92	10.72	15.47	12.01	10.12	8.06	7.92	7.03
			100	5.59	4.08	3.08	4.75	3.83	3.00	2.77	2.83	2.00
			1	56.80	40.73	33.24	49.06	38.10	31.65	26.91	24.93	22.70
		100	10	18.31	12.58	10.61	15.53	11.78	9.90	8.37	7.89	7.33
			100	5.92	3.96	3.05	4.69	3.88	3.00	2.92	2.77	2.00

Chapter 5

Integrating Order Scheduling with Packing and Delivery

5.1 Introduction

We consider a make-to-order supply chain consisting of one supplier (e.g. manufacturer) and one customer (e.g. retailer) where the supplier makes time sensitive products such as fashion apparel and customized high-tech products for the customer. At the beginning of the planning horizon, the customer places a set of orders with the supplier. The supplier needs to process these orders on a single dedicated production line, pack the completed orders to form delivery batches, and deliver the batches to the customer. Because of the time sensitivity of the products, the customer imposes a service requirement on the delivery timeliness of the orders she places with the supplier. The supplier needs to meet the imposed service requirement on the one hand, and minimize the total cost incurred for order processing and delivery on the other hand. Since the products are time sensitive, orders are delivered shortly after their completion and thus we assume that little inventory cost is incurred. The supplier's total cost is mainly contributed by production and distribution operations. The total production cost for a fixed set of orders is normally fixed and independent of the production schedule used. Therefore, the supplier needs to focus on the distribution cost when considering the total cost. Each order has a weight and the total weight of the orders that can be packed in each delivery batch must not exceed a capacity limit. Each delivery batch incurs a fixed distribution cost regardless of the total weight it carries.

The problem is to find jointly a schedule for order processing at the supplier, a way of packing completed orders to form delivery batches, and a delivery schedule from the supplier to the customer such that the total distribution cost is minimized subject to the constraint that a given customer service level is guaranteed. We consider two customer service constraints:

(a) Meeting the given deadlines of the orders.

(b) Requiring the average delivery lead time of the orders to be within a given threshold.

The problem with each of those constraints is studied separately. For ease of presentation, we call the problem with the service constraint (a) the deadline problem, and the problem with the service constraint (b) the lead time problem. For each problem, we consider the following three cases for the way an order can be produced and delivered:

(i) Non-splittable production and delivery: An order cannot be split in terms of production or delivery, i.e. it is not allowed to preempt the processing of an order and a finished order must be delivered in one batch.

(ii) Non-splittable production, but splittable delivery: An order cannot be split in terms of production, but can be split in terms of delivery, i.e. no processing preemption is allowed, but a finished order can be split into multiple parts delivered in multiple batches.

(iii) Splittable production and delivery: An order can be split in terms of both production and delivery, i.e. both processing preemption and delivery split of an order are allowed.

There are practical situations that justify each case. Splitting an order for either production or delivery may require extra setups. Furthermore, to ease order tracking and handling, the customer may require that an order be delivered wholly in one shipment. Hence, order splitting may not be desirable. On the other hand, allowing order splitting may improve service level or/and lower distribution cost. So depending on the requirements, the supplier may or may not be allowed to split production and/or delivery of orders. The problems we consider integrate order processing scheduling, finished order packing and batching, and order delivery scheduling.

In this chapter, we clarify the computational complexity of the various problems we consider by either proving that a problem is intractable (i.e. NP-hard) or providing an efficient exact algorithm for it. For the NP-hard problems, we design fast heuristics that are capable of generating near optimal solutions. We analyze the worst-case performance of each heuristic, and computationally evaluate their performance using randomly generated test instances. The remainder of the chapter is organized as follows. In Section 5.2, we define the problems and give some optimality properties. We then study the deadline problem in Section 5.3, and the lead time problem in Section 5.4, respectively. We look at three different cases, Case (i), Case (ii), and Case (iii), for both the deadline and the lead time versions. Both Cases (i) and (ii) of both problems are shown to be strongly NP-hard. Case (iii) of the deadline problem is solved optimally by a polynomial algorithm, whereas Case (iii) of the lead time problem is shown to be strongly NP-hard. A heuristic is proposed for each of the NP-hard problems. Column generation based approaches are proposed to find lower bounds of the objective values of those problems. Those lower bounds are then used to evaluate the performance of the heuristics. We provide two extensions in Section 5.5. Finally, we conclude the chapter in Section 5.6.

5.2 Problems and Preliminary Results

The supplier is given n orders, $N = \{1, \cdots, n\}$, at time 0, which are to be processed on a single production line. Each order $j \in N$ is associated with a given set of integer parameters: processing time p_j, deadline d_j, and weight w_j. Completed orders need to be packed to form batches and delivered to the customer in batches. The capacity of each delivery batch is b units, i.e. it can carry a subset of orders with a total weight of up to b units. The delivery time from the supplier to the customer is t and the delivery cost per batch is f, regardless of the total weight it carries. In a given schedule, we use C_j to denote the completion time of order j at the supplier, which is the time when order j is completed processing at the supplier, and D_j to denote the departure time from the supplier of the batch containing order j. Similarly, we use L_j to denote the delivery lead time of order j, which is the time when the order j is delivered to the customer. Clearly, $D_j \geq C_j$ and $L_j = D_j + t$.

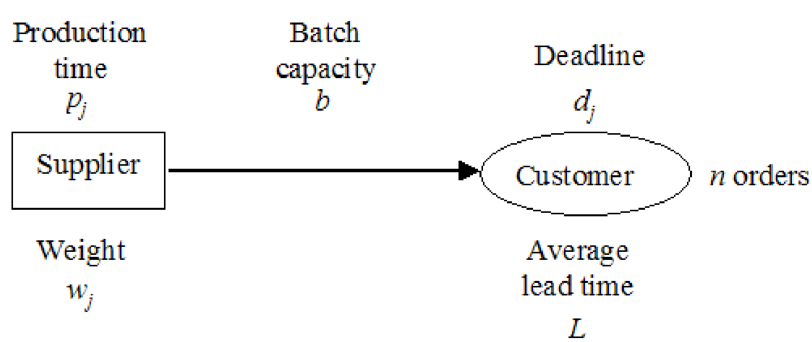

Figure 5.1: The supply chain

We consider three cases of the way an order can be processed and delivered, namely, Cases (i), (ii), and (iii) described in Section 5.1. In the case of splittable processing, it is allowed to split an order j into any number of parts and each part is allowed to have a non-integer processing time, as long as the total processing time spent on all the parts together is equal to p_j. In this case, the completion time of an order is the completion time of its last part, and the departure time of an order is the departure time of the batch containing the last part of the order. We assume that the weight of each split part is linearly proportional to its processing time, i.e. if a part of order j requires a processing time τ, then the its weight is $\tau w_j/p_j$, where the ratio w_j/p_j, which we call the filling-rate of order j, is the amount of weight corresponding to every unit of the processing time of order j. In the case of non-splittable delivery, an order j cannot be packed into a batch until its completion time C_j because the entire order has to be packed in the same batch. In the case of splittable delivery, any part of an order is available for delivery once this part is completed processing. The lead time of an order, if it is split into multiple parts delivered in multiple batches, is the time when the batch containing its last part is delivered to the customer.

We study two problems. The first one, called the deadline problem, is to minimize the number of delivery batches used subject to the service constraint (a) described in Section 5.1. That is, each order must be delivered to the customer no later than its deadline, or $L_j \leq d_j$ for $j \in N$. The second problem, called the lead time problem, is to minimize the number of delivery batches used subject to the service constraint (b) described in Section 5.1. That is, the average lead time of the orders should be no more than a given threshold L, or $(\sum_{j \in N} L_j)/n \leq L$. In our model, since all the orders are delivered to the same customer and the delivery time of any batch is the same, we can simplify these constraints as follows. We can redefine the deadline of each order j to be $d_j - t$ such that the service constraint (a) is equivalent to the constraint that the departure time D_j of order j is no later than the redefined deadline of j. Similarly, if we redefine the average lead time threshold to be $L - t$, then the service constraint (b) is equivalent to the constraint that the average departure time of the orders from the supplier is no more than the redefined threshold. Therefore, for ease of presentation, in the remainder of the chapter, we assume that the given deadlines of orders d_j's and the lead time threshold L have been redefined so that the constraint in the deadline problem becomes $D_j \leq d_j$ for $j \in N$, and that in the lead time problem becomes $(\sum_{j \in N} D_j)/n \leq L$.

To illustrate the differences between the three cases mentioned, consider the following problem instance for the deadline problem. We have five orders and the maximum batch size is given by $b = 19$. The five orders have the following parameters: $p_1 = p_2 = p_3 = 20$, $p_4 = p_5 = 10$; $d_1 = 23$, $d_2 = d_3 = 61$, $d_4 = d_5 = 80$; $w_1 = 17$, $w_2 = w_3 = w_4 = w_5 = 10$. We note that the sum of weights of all the orders is 57 and since the maximum batch size is 19, we need a minimum of three delivery batches. Below we give the optimal solutions for each case:

Case (i): We will have a total of five delivery batches in the optimal solution and each order will ship in a batch of its own. This is because each order has a weight more than $19/2 = 9.5$ and hence no two full orders can fit in one batch.

Case (ii): The optimal solution will have a total of four delivery batches. This is because, in order to obtain just three batches, we need all the batches to be full, including the first batch. The first batch has to get full by time $t = d_1 = 23$ and the only way for this to happen is to process order 4 or 5 immediately after processing order 1 (orders 2 and 3 have a lower filling-rate). But if this is done, then one of the orders 2 or 3 will miss their deadline since we cannot preempt the production of order 4 or 5 once it is started. Hence the first batch cannot be full, and we will need a total of four batches. One possible solution with four batches is as follows: first batch will have just the first order, second batch will have the second order and $\frac{9}{10}$th of the third order, the third batch will have the rest of the third order, and $\frac{1}{10}$th of the fourth order, and the fourth batch will have the rest. Hence the first batch will ship at time $t = 20$, the second will ship at $t = 58$, the third will ship at $t = 61$, and the fourth will ship at $t = 80$.

Case (iii): The following solution will meet all the deadlines with exactly three batches and hence it will be the optimal solution for Case (iii). Process order 1, $\frac{1}{10}$th of order 2, and $\frac{1}{10}$th of order 4 for the first batch. Process the remaining part of order 2 and then order 3 for the second batch. Process the remaining part of order 4, and ship it along with order 5 in the third batch. Thus the first batch will ship at time $t = 23$, the second will ship at $t = 61$, and the third will ship at $t = 80$.

The above example shows that the optimal solution will vary depending on the restrictions on production process or delivery batching. Next, we present some optimality properties that will be used in later sections. The first result holds for all the problems. It is fairly straightforward and hence we omit the proof. The second result holds for Cases (ii) and (iii) of both problems.

Lemma 17 There exists an optimal schedule for all the three cases of both problems where
(1) There is no inserted idle time between orders and partial orders processed at the supplier.
(2) The departure time of each batch is the time when processing of all the orders and partial orders in it are complete.
(3) All the orders and partial orders that are delivered in the same batch are processed consecutively at the supplier.

Lemma 18 There exists an optimal schedule for Case (ii) and Case (iii) of both problems where
(1) If a delivery batch contains a partial order which is not the first part of an order, then this batch is full.
(2) In each delivery batch containing partial orders, the weight of each partial order is an integer.

Proof (1) Suppose that there is a delivery batch B in a given optimal schedule π that is not full but contains a part of some order $j \in N$ which is not the first part of j. Let the total weight of the orders and parts of orders in this batch be v with $v < b$, and the weight of the part of order j contained in this batch be v_j with $v_j < w_j$. Clearly, the remaining part of order j has a weight $w_j - v_j$ and is contained in one or more batches delivered earlier than batch B. There are two cases as follows. Case 1: If $b - v \geq w_j - v_j$, we move all the earlier parts of order j to batch B. After that, the part of order j in batch B becomes either a whole order or the first part of it. Case 2: If $b - b < w_j - v_j$, we move a portion of the earlier parts of order j with a total weight $b - v$ to batch B. After that, batch B is full. In both cases, neither the completion time of each order nor the total number of batches used is increased, and thus the new schedule is still optimal.

(2) Suppose that in a given optimal schedule π, batch B is the last batch that violates this property, i.e. each partial order in every batch delivered later than B has an integer weight and there are partial orders in B with a non-integer weight. Let k be the number of partial orders in B with a non-integer weight. Let $j_1, \cdots, j_k$ denote those orders, and fractional values $v_{j_1}, \cdots, v_{j_k}$ denote their weights. For $h = 1, \cdots, k$, since the total weight of all the parts of order j_h delivered later than batch B has an integer total weight, one or more parts of j_h must be delivered earlier than batch B and the total weight of those parts is fractional. This means that the part of order j_h, for $h = 1, \cdots, k$, delivered in batch B is not the first part of the order. By Property (1), we can assume that batch B is full. Thus the total weight of the orders in B is b, an integer. Since a whole order has an integer weight and each partial order in B other than $j_1, \cdots, j_k$ has an integer weight, the total weight of the parts of orders $j_1, \cdots, j_k$ in B is an integer. Let α_h $(0 < \alpha < 1)$ be the fractional part of the weight of the part of order j_h in B, for $h = 1, \cdots, k$. Let $\rho = \alpha_1 + \cdots + \alpha_k$. Clearly, $\rho \leq k - 1$ and ρ is an integer. Let j_u, for some $u \in \{1, \cdots, k\}$, be the order with the highest filling-rate (recall that it is defined to be the ratio w_j/p_j) among the orders $j_1, \cdots, j_k$. Let

β be the total weight of the parts of orders $j_1, \cdots, j_{u-1}, j_{u+1}, \cdots, j_k$ contained in batches earlier than B. It can be easily verified that

$$\beta \geq \sum_{h=1}^{u-1}(1-\alpha_h) + \sum_{h=u+1}^{k}(1-\alpha_h) = (k-1) - \rho + \alpha_u \geq \alpha_u$$

We use a portion of order j_u contained in B with the weight α_u to exchange portion of the orders $j_1, \cdots, j_{u-1}, j_{u+1}, \cdots, j_k$ contained in batches earlier than B with an equal weight. Since order j_u has the highest filling-rate, this exchange will result in a schedule where each batch is delivered no later than in schedule π. After this exchange, the weight of order j_u in batch B is lowered by α_u units and thus becomes an integer. We can apply the same idea to the remaining orders with a fractional weight in batch B, $j_1, \cdots, j_{u-1}, j_{u+1}, \cdots, j_k$, and eventually every one of them will have an integer weight in B. ■

5.3 The Deadline Problem

In this section we consider the three cases of the deadline problem. Clearly, Case (i) of the problem is strongly NP-hard because the classical bin-packing problem, which is strongly NP-hard (Garey and Johnson 1979), is a special case of it with sufficiently large d_j for $j \in N$ such that the deadline constraint is always satisfied under any processing schedule. In Section 5.3.1, we clarify the computational complexity of the other two cases of the problem. We show that Case (ii) of the problem is strongly NP-hard, whereas there is a polynomial algorithm for Case (iii) of the problem. In Section 5.3.2, we propose two heuristics, one for Case (i), and one for Case (ii), respectively. We analyze the worst-case performance of the heuristics. In Section 5.3.3, we do a computational experiment on the heuristics. For Case (i), we propose a column generation approach to find a valid lower bound. Since Case (iii) is a relaxation of Case (ii), the optimal objective value of Case (iii) is a lower bound for Case (ii). We use these lower bounds to evaluate the performance of the heuristics computationally.

5.3.1 Solvability of Cases (ii) and (iii)

We first clarify the complexity of Case (ii) of the problem.

Theorem 18 The deadline problem with non-splittable production but splittable delivery is strongly NP-hard.

Proof We prove the theorem by a reduction from the 3-partition problem (3PP), a known strongly NP-hard problem (Garey and Johnson 1979).

3PP: Given a set of $3n$ elements, $A = \{1, \cdots, 3n\}$, a positive integer H, and a positive integer size a_i for each element $i \in A$ with $H/4 < a_i < H/2$, such that the total size of elements in A is exactly nH, can A be partitioned into n disjoint subsets $A_1, \cdots, A_n$ such that each A_i contains exactly 3 elements of A and has a total size equal to H?

We construct the following instance for our problem where there are $5n$ orders consisting of three types:

Type 1: n orders $N_1 = \{1_1, \cdots, 1_n\}$ with parameters: $p_{1_i} = 9H, w_{1_i} = 9H, d_{1_i} = 30H(i-1) + 10H$, for $i = 1, \cdots, n$.

Type 2: n orders $N_2 = \{2_1, \cdots, 2_n\}$ with parameters: $p_{2_i} = 20H, w_{2_i} = 10H, d_{2_i} = 30Hi$, for $i = 1, \cdots, n$.

Type 3: $3n$ orders $N_3 = \{3_1, \cdots, 3_{3n}\}$ with parameters: $p_{3_i} = a_i, w_{3_i} = a_i, d_{3_i} = 30Hn$, for $i = 1, \cdots, 3n$.

Maximum delivery batch size: $b = 10H$.

Threshold value for the objective function: $2n$ delivery batches.

(If part) If there is a solution to 3PP, we construct a solution to the above instance of our problem as follows. Process the orders in the sequence $(1_1, 3_{A_1}, 2_1, 1_2, 3_{A_2}, 2_2, \cdots, 1_n, 3_{A_n}, 2_n)$, where the subset $3_{A_i} = \{3_j \mid j \in A_i\}$, for $i = 1, \cdots, n$. Make a total of $2n$ deliveries where, for $k = 1, \cdots, n$, the $(2k-1)$th delivery covers order 1_k and the 3 orders of 3_{A_k}, and the $(2k)$th delivery covers order 2_k. We can see that in this solution, all the delivery batches are full, and all the orders satisfy their deadlines.

(Only If part) Recall that we defined the filling-rate of an order j in Section 5.2 as the ratio of its weight to its processing time w_j/p_j. We denote the filling-rate of order j to be f_j. Clearly, $f_j = 1$ for $j \in N_1 \cup N_3$ and $f_j = 0.5$ for $j \in N_2$. Since we cannot have more than $2n$ delivery batches and since the total weight of orders is $20nH$, there are exactly $2n$ delivery batches and all the delivery batches are full in any optimal solution. The minimum total processing time required by the orders and partial orders in a full delivery batch is $\min\{10H/f_j, \mid j \in N_1 \cup N_2 \cup N_3\} = 10H$.

Let t_i denote the time at which the ith delivery batch becomes full (i.e. t_i is the departure time of this batch). Let S_j denote the time at which order j begins processing. Let us consider the first delivery batch. Since $d_{1_1} = 10H$ and since it takes a minimum of $10H$ time units to fill a delivery batch completely, order 1_1 has to be in the first delivery batch. Hence $t_1 = 10H$. Since $w_{1_1} = 9H$, there is some empty space in the first delivery batch after processing order 1_1. This has to be filled with one or more orders or partial orders because every delivery batch must be full. If a second Type-1 order is processed after order 1_1, since $p_{1_i} = 9H$, it will be completely processed at time$18H$, which implies that $S_{2_1} \geq 18H$, and hence $C_{2_1} = S_{2_1} + p_{2_1} \geq 38H > d_{2_1}$. So putting a second Type-1 order after order 1_1 will lead us to an infeasible solution. If we put any Type-2 order 2_i after order 1_1, it will take $H/f_{2_i} = 2H$ units of time to fill up the first delivery batch. This means that $t_1 = 9H + 2H = 11H > d_{1_1}$, again resulting in an infeasible solution. So the only option is to process one or more Type-3 orders to go with order 1_1 in the first delivery batch to be shipped at time $t_1 = 10H$. We denote by ϕ the set of Type-3 orders in the first delivery batch either partially or entirely along with order 1_1. There are two cases to consider:

Case 1: $\sum_{j\in\phi} p_j > H$. In this case, the first Type-2 order will have a starting time $S_{2_1} > 9H + H = 10H$ and hence $C_{2_1} = S_{2_1} + p_{2_1} > 30H = d_{2_1}$. Hence, this case is not feasible.

Case 2: $\sum_{j\in\phi} p_j < H$. Denote $p_\phi = \sum_{j\in\phi} p_j$. In this case, the total weight of the orders in ϕ is $\sum_{j\in\phi} w_j = p_\phi$ and the delivery batch will not be full with just these orders and order 1_1. There will be an empty space equivalent to $H - p_\phi$ units of weight and $H - p_\phi$ units of time to go before the departure time of the first delivery $t_1 = 10H$. We will have to add a part of a Type-1 or/and a part of a Type-2 order to make the first delivery batch full. If we add a part of a Type-1 order, then by a similar argument used earlier, this will result in the first Type-2 order getting completed later than its deadline. So we will have to add a part of Type-2 order instead. This will take $(H - p_\phi)/0.5 = 2H - 2p_\phi$ time units to fill up the spare space of the first delivery batch. So t_1 would become $9H + p_\phi + 2H - 2p_\phi = 11H - p_\phi > 10H = d_{1_1}$, resulting in an infeasible solution.

From the above cases, it is clear that $\sum_{j\in\phi} p_j = H$ and the first batch contains entire order 1_1 and entire orders in ϕ. Once the first batch is full at time $t_1 = 10H$, we have to start processing order 2_1 immediately so that $C_{2_1} = t_1 + p_{2_1} = 30H = d_{2_1}$. Since $w_{2_1} = 10H$, this order will be shipped individually in the second delivery batch at $t_2 = 30H$. And once this is done, we look at the third delivery batch and we are in a situation that is exactly the same as the first delivery batch. Hence, continuing this, we can see that each odd numbered batch will contain one Type-1 order and three Type-3 orders that add to a weight of H, and each even numbered batch will contain one Type-2 order shipped independently. Define subsets

A_k of A, for $k = 1, \cdots, n$, such that the three Type-3 orders in the $(2k-1)$th delivery batch are $\{3_i \mid i \in A_k\}$. Then $A_1, \cdots, A_n$ form a partition of A that solves 3PP. ■

Next, we consider Case (iii) of the deadline problem. For this problem, the following polynomial algorithm finds an optimal solution.

Algorithm A1

Step 0: Schedule orders from time $P = \sum_{j \in N} p_j$ to 0 backwards. Set the current time $t = P$.

Step 1: Find all the unscheduled orders (including partial orders) which have a deadline greater than or equal to t. Let this order set be S. If S is empty, then the problem is infeasible. Otherwise, let the total processing time and weight of these orders be P_S and W_S, respectively.

Step 2: If $W_S \leq b$, process these orders in any sequence from time $t - P_S$ to time t, and ship them in one batch at time t. Update $t = t - P_S$. Go to Step 1.

Step 3: If $W_S > b$, let $\tau = t$, and sort the orders in the non-decreasing sequence of their filling-rates w_j/p_j. Let this sequence of orders be denoted as $[1], [2], \cdots, [h]$, where h is the number of orders in S. Let u be such that $\sum_{j=1}^{u} w_{[j]} \leq b$ and $\sum_{j=1}^{u+1} w_{[j]} > b$. There are two cases as follows.

If $\sum_{j=1}^{u} w_{[j]} = b$, then process the first u orders $[1], \cdots, [u]$, from time t backwards. Let $t = t - \sum_{j=1}^{u} p_{[j]}$.

If $\sum_{j=1}^{u} w_{[j]} < b$, define $\alpha = (b - \sum_{j=1}^{u} w_{[j]})/w_{[u+1]}$, and process the first u orders $[1], \cdots, [u]$, and $\alpha \times 100\%$ of order $[u+1]$, from time t backwards. Let $t = t - \sum_{j=1}^{u} p_{[j]} - p_{[u+1]}\alpha$. Reset the remaining processing time and weight of order $[u+1]$ as $p_{[u+1]} = p_{[u+1]}(1-\alpha)$ and $w_{[u+1]} = w_{[u+1]}(1-\alpha)$ respectively.

Deliver all the orders and partial orders that have been processed in this step in one shipment at time τ. Go to Step 1.

Theorem 19 Algorithm A1 finds an optimal solution to the deadline problem with splittable production and delivery in $O(n^2 \log n)$ time.

Proof We first estimate the computational time of the algorithm. Each iteration of the algorithm (running Steps 1, and 2 or 3, once) schedules a new batch of orders selected from a set S, where S includes all eligible orders and partial orders. Since the weight of each order is no more than the batch capacity, if S is not empty, then in either Step 2 or Step 3, at least one order or partial order $k \in S$ is 100% covered by this batch. In case k is a partial order, clearly its remaining part is covered in a later batch (which has been generated by the algorithm in an earlier iteration since the algorithm schedules orders backwards). This means that order k is completely covered in batches generated in this and earlier iterations. So we can conclude that after each iteration of the algorithm, at least one new order is completely covered. Since there are n orders, the algorithm stops after at most n iterations. In each iteration, we may need to arrange at most n orders and partial orders in the non-decreasing sequence of w_j/p_j. This has a time requirement of $O(n \log n)$. So the overall computational complexity of the algorithm is $O(n^2 \log n)$.

To show that the solution π generated by Algorithm A1 is optimal, we show that any optimal solution π^* can be transformed to π without increasing the number of delivery batches. Suppose that for some integer $h \geq 0$, the kth last batch in π^* is exactly the same (i.e. contains the same set of orders, each with the same weight or partial weight) as the kth last batch in π, for $k = 0, \cdots, h$, but the $(h+1)$th last batch in π^*, denoted as $B_{h+1}(\pi^*)$, is different from the $(h+1)$th last batch in π, denoted as $B_{h+1}(\pi)$. We can easily show that the set S obtained in the $(h+1)$th iteration in Step 1 of algorithm A1 is the same for both

π and π^*. Define W_s and P_s to be the total processing time and weight of the orders in S, respectively. There are two cases as follows.

Case 1: If $W_S \leq b$, then by the algorithm, all the orders of S are included in $B_{h+1}(\pi)$. Since $B_{h+1}(\pi^*) \neq B_{h+1}(\pi)$, and since S contains all the eligible orders and partial orders that may be delivered in $B_{h+1}(\pi^*)$, there exists some order $j \in S$ which is not entirely included in $B_{h+1}(\pi^*)$. We can move the remaining part of order j, which must be scheduled in an earlier batch in π^*, to batch $B_{h+1}(\pi^*)$. This will not increase the departure time of any batch, and hence is feasible. Furthermore, the number of batches is not increased. We can continue this procedure until $B_{h+1}(\pi^*) = B_{h+1}(\pi)$.

Case 2: If $W_S > b$, then by Step 3 of the algorithm, the first u orders and a part of the $(u+1)$th order with the lowest filling-rates among all the orders in S are included in $B_{h+1}(\pi)$, and this is a full batch. If batch $B_{h+1}(\pi^*)$ is not full, then we can move some orders or/and parts of some orders of S scheduled in earlier batches in π^* to batch $B_{h+1}(\pi^*)$ to make it a full batch without increasing the departure time of any batch or the total number of batches. Now we can assume that $B_{h+1}(\pi^*)$ is also a full batch, but different than $B_{h+1}(\pi)$. Suppose that, for some $1 \leq k \leq u$, $B_{h+1}(\pi^*)$ contains the first k orders $[1], \cdots, [k]$ in exactly the same amount as it is contained in $B_{h+1}(\pi)$, but contains only a portion (which can be 0%) of order $[k+1]$ that is contained in $B_{h+1}(\pi)$. Then there must exist one or more orders, denoted by set Q, in $B_{h+1}(\pi^*)$ that have a higher filling-rate than order $[k+1]$. We can swap part of order $[k+1]$ scheduled in earlier batches with part of orders of Q in π^* such that after swapping the amount of order $[k+1]$ in $B_{h+1}(\pi^*)$ becomes exactly the same as that in batch $B_{h+1}(\pi)$. This will not increase the departure time of any batch or the total number of batches. We can continue such a swapping procedure until $B_{h+1}(\pi^*) = B_{h+1}(\pi)$.

■

5.3.2 Heuristics for Cases (i) and (ii)

In this section, we propose a heuristic for Case (i) and Case (ii) of the deadline problem, respectively.

In the heuristic for Case (i), a delivery batch departs at the time corresponding to the earliest deadline among all the orders in the delivery batch. Thus a delivery batch is said to be available at time t if and only if t is not greater than the departure time of the delivery batch.

Heuristic H1

Step 0: Reindex orders in their EDD sequence. Process the orders in the sequence of their order indices without idle time. Let $j = 1$.

Step 1: Consider order j. Fit it in the first feasible and available delivery batch. If the order cannot be fit into any of the available delivery batches, open a new delivery batch.

Step 2: Set $j = j + 1$. If $j \leq n$, go to Step 1. Else STOP.

Theorem 20 The worst-case performance ratio of Heuristic H1 for the deadline problem with non-splittable production and delivery is bounded by 3.

Proof Note that in H1, the delivery batches are delivered in the order in which they are opened and a new delivery batch can be opened due to one of the following reasons:

i) The order under consideration does not fit into any available delivery batches.

ii) All earlier delivery batches have been delivered and hence there are no delivery batches available.

We divide the delivery batches into two categories: Category S contains those delivery batches that were opened because of (i), category U contains all the other delivery batches.

In this arrangement, we will have alternating series of S and U type delivery batches (e.g. one U type delivery batch followed by three S type delivery batches followed by two U type followed by two S type). If a particular series of S type delivery batches has an odd number of members, put the last delivery batch in the preceding U type to that series. Thus S will have even numbered delivery batches in each segment. Let s and u be the total number of such delivery batches in S and U respectively. We argue that the optimal solution will have at least $\max\left\{\lceil\frac{s}{2}\rceil, u\right\}$ delivery batches. If this is the case, then the worst case-ratio for our heuristic is $\frac{s+u}{\max\left\{\lceil\frac{s}{2}\rceil, u\right\}} \leq 3$.

It can be easily shown that all the orders in S add upto a total weight of at least $\frac{sb}{2}$. So we need at least $\lceil\frac{s}{2}\rceil$ delivery batches for those orders. Let us look at the U type delivery batches. Let the sum of processing times of all the orders up to the first order in the ith U type delivery batch be denoted as P_i. Let the delivery batch be denoted as U_i, and the index for the first order in U_i be denoted as x_i. Thus P_i is the sum of processing times of all the orders with an index k such that $k \leq x_i$. Consider all the orders up to x_2. The sum of their processing times is P_2. But since at P_2, U_1 has already left, we have $P_2 \geq d_{x_1}$, the deadline corresponding to the first order in the first U category delivery batch. This shows that whichever sequence the production is carried out, at least one order from $\{1, \cdots, x_1, \cdots, x_2\}$ gets ready after d_{x_1} and that order has to be shipped in a different delivery batch after d_{x_1}. Let this particular order in the optimal solution be denoted as y_2. Note also that d_{x_2} is the highest deadline among all the orders in $\{x_1, \cdots, x_2\}$ and hence $d_{x_2} \geq d_{y_2}$. Now consider x_3. Since at P_3, U_2 has already left, we have $P_3 \geq d_{x_2}$, the deadline corresponding to the first order in U_2. This means that whichever sequence the production is carried out, at least one order from $\{1, \cdots, x_1, \cdots, x_2, \cdots, x_3\}$ gets ready after d_{x_2} (and d_{y_2}) and that order has to be shipped in a delivery batch that is different from the delivery batch containing x_2 or y_2. Let this order in the optimal solution be y_3. Proceeding like this, we can see that corresponding to each U category delivery batch, there should be a separate shipment. This shows that minimum number of shipments required is bound from below by u.

Therefore, the minimum number of delivery batches for the optimal solution for our problem is: $\max\left\{\lceil \frac{s}{2} \rceil, u\right\}$. Thus the worst-case ratio is: $\frac{s+u}{\max\left\{\lceil \frac{s}{2} \rceil, u\right\}}$. If $u \geq \lceil \frac{s}{2} \rceil$, it implies that $s \leq 2u$. On the other hand, when $u < \lceil \frac{s}{2} \rceil$, we have $s \leq 2\lceil \frac{s}{2} \rceil$. Therefore, the worst-case ratio is bound by 3. ■

The heuristic for Case (ii) generates a feasible solution by modifying the solution generated by Algorithm A1 for Case (iii) of the problem given in Section 5.3. Note that the solution generated by Algorithm A1 allows production preemption and hence is generally not feasible to Case (ii). Below we give the heuristic and its worst-case complexity.

Heuristic H2

Step 0: Run Algorithm A1. Let the processing sequence of the orders in the solution generated by Algorithm A1 be π^0. If there is no production preemption in this solution, then STOP, and this solution is optimal to Case (ii).

Step 1: Examine the processing sequence π^0 from time $P = \sum_{j \in N} p_j$ to 0 backwards. Whenever there is a partial order j, move all the other parts of this order (which must be scheduled earlier) to the position immediately before this partial order so that the whole order j is processed without preemption after this rearrangement. Let the new processing sequence of the orders be π.

Step 2: Given the processing sequence of the orders π (where there is no production preemption), find a packing and delivery schedule by the following procedure:

Step 2.0: Reindex the orders such that $\pi = (n, n-1, \cdots, 1)$. Consider orders in π from time $P = \sum_{j \in N} p_j$ to 0 backwards. Set the current time $t = P$, and the current order $k = 1$. Define the remaining processing time and weight of the current order to be $p'_1 = p_1$ and $w'_1 = w_1$, respectively. No order has been assigned to a batch yet.

Step 2.1: If every order or partial order has been assigned to a batch, STOP. Find the maximum integer h with $h \geq k$ such that all the orders $k, \cdots, h$ in π have a deadline greater than or equal to t. Let $P' = p'_k + \sum_{j=k+1}^{h} p_j$ and $W' = w'_k + \sum_{j=k+1}^{h} w_j$ be the total processing time and weight of the remaining part of order k and the entire orders $k+1, \cdots, h$, respectively. Consider the next batch.

Step 2.2: If $W' \leq b$, ship the remaining part of order k and the entire orders $k+1, \cdots, h$ in one batch at time t. Update $t = t - P'$, $k = h + 1$, $p'_k = p_{h+1}$ and $w'_k = w_{h+1}$. Go to Step 2.1.

Step 2.3: If $W' > b$, find r with $k \leq r < h$ such that $w'_k + \sum_{j=k+1}^{r} w_j \leq b$ and $w'_k + \sum_{j=k+1}^{r+1} w_j > b$. Define $\alpha = (b - w'_k - \sum_{j=k+1}^{r} w_j)/w_{r+1}$. Ship the following orders in one batch at time t: the remaining part of order k, the entire orders $k+1, \cdots, r$, and $\alpha \times 100\%$ of order $r+1$. Update $t = t - p'_k - \sum_{j=k+1}^{r} p_j - \alpha p_{r+1}$, $k = r+1$, $p'_k = (1-\alpha) p_{r+1}$ and $w'_k = (1-\alpha) w_{r+1}$. Go to Step 2.1.

The processing sequence π^0 generated in Step 0 of the heuristic may have production preemptions. However, the new sequence π generated in Step 1 does not have any production preemption. Step 2 forms delivery batches based on the processing sequence π such that a maximum amount of weight possible is included in each batch.

Theorem 21 The worst-case performance ratio of Heuristic H2 for the deadline problem with non-splittable production but splittable delivery is bounded by 2.

Proof Let x^0, x^{H2}, and x^* be, respectively, the number of batches in the solution generated by Algorithm A1 (for Case (iii)), that in the solution generated by Heuristic H2 (for Case (ii)), and that in the optimal solution of Case (ii). Since Case (iii) is a relaxation of Case (ii), we have:

$$x^* \geq x^0 \tag{5.1}$$

In Heuristic H2, given the processing sequence of orders π, Step 2 forms delivery batches backwards from time P to time 0 and a maximum amount of weight is assigned to each batch. It can be easily shown that

Claim 1: The number of batches generated in Step 2 is the minimum possible given the processing sequence π and the fact that the batches formed satisfy the property that orders delivered together in a batch are processed consecutively in π.

Let $B_1^0, \cdots, B_{x^0}^0$ denote the x^0 batches in the solution generated by Algorithm A1, where a batch with a smaller index is delivered earlier. It can be seen that there is at most one partial order in each batch B_k^0, for $k = 1, \cdots, x^0$, whose last part is included in this batch. For $k = 1, \cdots, x^0$, let $[k]$ denote the order index of such partial order in B_k^0, where $[k]$ is a null order (with 0 processing time and weight) if no such partial order exists in B_k^0. Clearly, $[1]$ must be null because it is not possible for batch B_1^0 to contain a partial order whose last part is included in it. In Step 1 of the heuristic, for every such partial order, all the other parts of this order which are processed earlier are moved to the position immediately before this partial order. This means that the processing sequence π generated in Step 1 of the heuristic has the following structure: $(B_1^0 \setminus R_1, [2], B_2^0 \setminus R_2, [3], \cdots, B_{x^0-1}^0 \setminus R_{x^0-1}, [x^0], B_{x^0}^0 \setminus R_{x^0})$, where $R_k = \{[k], \cdots, [x^0]\}$.

Now we generate a new packing and delivery schedule based on the processing sequence π as follows: Deliver all the orders in $B_1^0 \setminus R_1$ in one batch, and for $k = 2, \cdots, x^0$, deliver order $[k]$ in one batch if $[k]$ is not a null order, and all the orders in $B_k^0 \setminus R_k$ in one batch.

The departure time of each batch is set to be the completion time of the last order in the batch. It can be easily seen that both the departure time of the batch containing the orders of $B_k^0 \setminus R_k$ and that of the batch containing a single order $[k]$ in this new schedule are no later than that of batch B_k^0 in the schedule generated by Algorithm A1. Thus, the deadline of each order is met in this new schedule. Also, it can be seen that there is no production preemption in this new schedule. Therefore, this new schedule is feasible for Case (ii). Furthermore, this new schedule is based on the processing sequence π and the orders delivered in each batch are processed consecutively. Since there are at most $2x^0$ batches in this new schedule, by Claim 1, we have:

$$x^{H2} \leq 2x^0 \tag{5.2}$$

By (5.1) and (5.2), we have: $x^{H2} \leq 2x^*$. This shows the theorem. ■

5.3.3 Computational Experiment

In this section, we test the performance of the heuristics proposed for Cases (i) and (ii) of the deadline problem. We use column generation to obtain a lower bound for the problem under Case (i). We use the optimal objective value of Case (iii) of the problem obtained by Algorithm A1 as a lower bound for Case (ii). These lower bounds are used to evaluate the performance of Heuristics H1 and H2 computationally based on randomly generated test instances.

Column generation for Case (i)

We first given an integer programming formulation for Case (i) of the problem. Let Ω denote the set of all feasible delivery batch configurations. A batch $\omega \in \Omega$ is defined based on the following parameters: (i) the departure time the batch, and (ii) the set of orders in the batch. Let x_ω be a binary variable that is 1 if batch ω is part of the final solution, and

0 otherwise. Since the sequence in which the orders in a batch are processed does not affect the solution, we assume that in any batch, the orders are processed in the reverse sequence of their indices (this makes the description of DP1, to be given later, simpler). We assume that the orders have been indexed in their EDD sequence with ties in due dates broken based on the increasing order of their processing times. We define the following parameters:

$P = \sum_{j \in N} p_j$ = total processing time of the orders

$a_{j\omega} = 1$ if order j is covered in batch ω and 0 otherwise

$b_{t\omega} = 1$ if there is an order in batch ω which is processed over a period of time that covers interval $[t, t+1]$, and 0 otherwise

We have the following formulation:

$$[\text{IP1}] \quad \min \sum_{\omega \in \Omega} x_\omega \tag{5.3}$$

Subject to:

$$\sum_{\omega \in \Omega} a_{j\omega} x_\omega = 1 \qquad \forall j \in N \tag{5.4}$$

$$\sum_{\omega \in \Omega} b_{t\omega} x_\omega = 1 \qquad \forall t \in \{0, \cdots, P-1\} \tag{5.5}$$

$$x_\omega \in \{0, 1\} \qquad \forall \omega \in \Omega \tag{5.6}$$

In [IP1], the objective function minimizes the number of batches. Constraint (5.4) ensures that each order is covered in the final schedule. Constraint (5.5) ensures that every time slot in the interval $[0, P]$ is covered exactly once. We denote the LP relaxation of [IP1] as [LP1], where constraint (5.6) is replaced by $x_\omega \geq 0$. Clearly, the optimal solution value for [LP1] is lower bound for [IP1] and hence for Case (i) of our problem.

Since the number of batches in Ω can be extremely large, we do not solve [LP1] directly. Instead, we use a column generation approach to generate necessary batches only. In the column generation approach, we solve a master problem with a subset of columns, and then solve a subproblem to obtain new columns with a negative reduced cost. We iterate between the master problem and the subproblem till no more new columns with negative reduced costs are found.

An initial set of columns can be generated by processing all the orders in the sequence of their deadlines and shipping each order independently. We use π_j and σ_t to denote the dual variable value corresponding to the constraint set (5.4) and (5.5) respectively. Then the reduced cost r_ω of the column corresponding to $\omega \in \Omega$ is given by: $r_\omega = 1 - \sum_{j \in \omega} \pi_j - \sum_{t=t_{\omega s}}^{t_{\omega e}-1} \sigma_t$, where the orders of batch ω are processed in the time interval $[t_{\omega s}, t_{\omega e}]$. The subproblem in each iteration of the column generation algorithm is to find a batch $\omega \in \Omega$ with the minimum r_ω.

In the following, we develop a dynamic programming (DP) algorithm for solving the subproblem.

Algorithm DP1

Reindex the orders in their EDD order.

Define the value function $F_e(j, t, w)$ as the minimum reduced cost of a batch consisting of a subset of the first j orders $\{1, \cdots, j\}$, given that the batch departs at time e, all the orders in the batch have a deadline at or later than time e, the total weight of the orders in the batch is w, and the total processing time of the orders in the batch is t, where $0 \leq e \leq P$, $1 \leq j \leq n$, $0 \leq t \leq e$, $0 \leq w \leq b$.

Initial Values

$F_e(0, 0, 0) = 1$ for $0 \leq e \leq P$.

$F_e(j, t, w) = \infty$ for any state (e, j, t, w) violating any of the following conditions: $0 \leq e \leq P$, $1 \leq j \leq n$, $0 \leq t \leq e$, $0 \leq w \leq b$.

Recursive Relations

$$F_e(j,t,w) = \begin{cases} \min\left\{F_e(j-1,t,w), \quad F_e(j-1,t-p_j,w-w_j) - \pi_j - \sum_{\tau\in[e-t,e-(t-p_j+1)]}\sigma_\tau\right\} \\ \hfill \text{if } d_j \geq e \\ F_e(j-1,t,w), \quad \text{otherwise} \end{cases}$$

Optimal Solution is found by solving:

$\min\{F_e(j,t,w)|1 \leq j \leq n, p_{\min} \leq e \leq P, p_{\min} \leq t \leq e, 0 \leq w \leq b\}$, where $p_{\min}$ is the minimum processing time among all the orders.

Lemma 19 Algorithm DP1 solves the problem of finding the minimum reduced cost optimally in $O(nbP^2)$ time.

Proof The term $-\pi_j - \sum_{\tau\in[e-t,e-(t-p_j+1)]}\sigma_\tau$ in the recursive relation is the contribution made by order j to the reduced cost if it is started processing at time $e-t$. This enumerates all possible orders that can be placed in an existing partially filled batch. So the optimality is guaranteed. Since e and t could take $(P+1)$ different values each, j could take n different values, and w could take $(b+1)$ different values, the time complexity of the algorithm is $O(nbP^2)$. ■

Computational results

For H1, in addition to trying the first fit approach for the orders, we also try the best fit approach when an order is put in a batch among the available batches that gives the minimum amount of the leftover space. We take the better of the two solutions and compare it with the lower bound obtained through column generation described in the previous subsection. For Case (ii), since Case (iii) is a relaxed version of Case (ii), the optimal objective value of Case (iii) is a lower bound of the optimal objective value of Case (ii).

Therefore, we use this lower bound for Case (ii). We generate random test problems with the following parameter configurations:

a. Number of orders $n \in \{20, 40, 60\}$

b. Delivery batch capacity, $b = 20$

c. Order processing times are independently generated from a uniform distribution $U[1, 20]$

d. Order sizes are independently generated from a random distribution $U[1, xb]$ where $x \in \{0.5, 0.75, 1\}$

f. Three cases of deadlines are considered: High, Medium, and Low. We define these based on the average number of delivery batches resulting out of the deadline settings. Let $h_l = \lceil \sum_{j \in N} w_j / b \rceil$, and $h_u = n$. Clearly, h_l and h_u are lower and bounds on the number of delivery batches required to deliver all the orders. Initial values of order deadlines are randomly generated from the distribution $U[1, \alpha P]$, where α is a parameter we use to control the tightness of the deadlines. We will specify later how the value of α is set. Given α, let d'_j be the initial deadline of order j generated. The actual value of the deadline of order j is set to be $\max\{d'_j, P_j\}$, where P_j is the total processing time of order j and the orders sequenced before j in an EDD sequence of the orders based on their initial deadlines. Clearly, P_j is the minimum possible value for the deadline of order j in order for the problem to be feasible. The value of α is set so that the average number of delivery batches in the solutions generated by the heuristic is approximately equal to $h_l + 0.2(h_u - h_l)$ for the High case, $h_l + 0.5(h_u - h_l)$ for the Medium case, and $h_l + 0.8(h_u - h_l)$ for the Low case, respectively.

We test five randomly generated problem instances for each combination of the parameters. Since we have a total of $3 \times 3 \times 3 = 27$ different combination of parameters, we get a total of 135 different problem instances. For each instance, we calculate the relative gap

$\frac{Z^H - Z^{LB}}{Z^{LB}} \times 100\%$ where Z^H is the solution given by a heuristic H1 or H2 and Z^{LB} is a lower bound which is generated by the column generation approach for Case (i), and Algorithm A1 applied to Case (iii) for Case (ii).

Table 5.1 gives the results for Case (i). The overall average and maximum gaps are 5.27% and 22.22% respectively. The average gaps for the High, Medium, and Low deadline cases are 1.88%, 6.04%, and 7.89% respectively. The heuristic performs the best under tight deadlines because when the deadlines are tight, most orders get shipped independently and hence most of the delivery batches are not full even in the optimal solution. When the deadlines are loose, delivery batches consolidate multiple orders and the performance of the heuristic is bound by the first-fit or the best-fit heuristic that is used to batch the orders. The average gaps for the 20, 40, and 60 order instances are 5.07%, 6.17%, and 4.57% respectively, and for the three different average order sizes are 4.83%, 5.50%, and 5.47% respectively. Hence we do not observe any trend in both cases.

For Case (ii), the results are given in Table 5.2. The overall average gap is 0.69% while the maximum gap is 12.50%. There were 19 out of 135 instances for which the heuristic solution was not equal to the lower bound (i.e. was not optimal). But there was only one problem instance where the heuristic solution had two extra bins, in all the other cases, the difference was just one. The average gaps for the High, Medium, and Low deadline cases are 0.13%, 0.71%, and 1.23% respectively. Again, this is because probability for partial orders and hence production preemption is higher when the deadlines are loose rather than tight. The average gaps for the 20, 40, and 60 order instances are 0.58%, 0.81%, and 0.67% respectively, and for the three different average order sizes are 0.56%, 0.59%, and 0.91% respectively. While we do not observe any trend with respect to the number of orders, the gaps seem to increase when the average size of the orders is increased. This may be due to the fact that the impact of having a partial order in a batch (and hence possibility of preemption) is higher when the order sizes are large compared to when they are small.

5.4 The Lead Time Problem

In this section we consider the three cases of the lead time problem. Clearly, Case (i) of the problem is strongly NP-hard because the classical bin-packing problem, which is strongly NP-hard (Garey and Johnson 1979), is a special case of it if we set the threshold of the average lead time L sufficiently large such that the average lead time constraint is always satisfied under any processing schedule. In Section 5.4.1, we show that both Case (ii) and Case (iii) of the problem are strongly NP-hard. In Section 5.4.2, we propose a heuristic for each case of the problem. We analyze the worst-case performance of the heuristics. In Section 5.4.3, we do a computational experiment on the heuristics. We use a lower bound generated by a column generation approach for each of the three problems. It is shown that the heuristics are capable of generating near optimal solutions for all the cases of the problem.

5.4.1 Solvability of Cases (ii) and (iii)

In this section we show that both Case (ii) and Case (iii) of the lead time problem are strongly NP-hard. We first give a result which will be used later in the NP-hardness proof.

Lemma 20 Given any positive integer k and any k constants $E_1, \cdots, E_k$ with $E_1 < E_2 < \cdots < E_k$, the following holds: $\sum_{j=1}^{k} E_j y_j \geq 3 \sum_{j=1}^{k} E_j$ for any non-negative integers $y_1, \cdots, y_k$ satisfying: $\sum_{j=i}^{k} y_j \geq 3(k-i+1)$ for all $i = 1, \cdots, k$, and $\sum_{j=1}^{k} y_j = 3k$.

Proof We show this by induction on k. The result is clearly true when $k = 1$. Assuming that the result holds when $k = u$ for some positive integer $u \geq 1$, we need to prove that the result holds when $k = u + 1$. When $k = u + 1$, given any non-negative integers $y_1, \cdots, y_k$

satisfying: $\sum_{j=i}^{k} y_j \geq 3(k-i+1)$ for all $i = 1, \cdots, k$, and $\sum_{j=1}^{k} y_j = 3k$, there are two cases to consider: (i) $y_{u+1} = 3$; (ii) $y_{u+1} \geq 4$.

In case (i), $y_1, \cdots, y_u$ satisfy: $\sum_{j=i}^{u} y_j \geq 3(u-i+1)$ for all $i = 1, \cdots, u$, and $\sum_{j=1}^{u} y_j = 3u$. Thus by the induction assumption, $\sum_{j=1}^{u} E_j y_j \geq 3\sum_{j=1}^{u} E_j$, which, together with the fact that $y_{u+1} = 3$, implies that $\sum_{j=1}^{u+1} E_j y_j \geq 3\sum_{j=1}^{u+1} E_j$.

In case (ii), let $y'_u = y_u + y_{u+1} - 3$, and $y'_j = y_j$ for $j = 1, \cdots, u-1$. Clearly, $y'_1, \cdots, y'_u$ satisfy: $\sum_{j=i}^{u} y'_j \geq 3(u-i+1)$ for all $i = 1, \cdots, u$, and $\sum_{j=1}^{u} y'_j = 3u$. We have

$$\begin{aligned}
\sum_{j=1}^{u+1} E_j y_j &= 3E_{u+1} + (y_{u+1} - 3)E_{u+1} + \sum_{j=1}^{u} y_j E_j \\
&\geq 3E_{u+1} + \sum_{j=1}^{u} y'_j E_j \quad (\text{since } E_{u+1} > E_u) \\
&\geq 3E_{u+1} + 3\sum_{j=1}^{u} E_j \quad (\text{by induction assumption}) \\
&= 3\sum_{j=1}^{u+1} E_j
\end{aligned}$$

The above shows that the result holds when $k = u+1$ under both cases of y_{u+1}. This completes the proof. ■

Theorem 22 The lead time problem with non-splittable production but splittable delivery is strongly NP-hard.

Proof We prove this by a reduction from the 3-partition problem (3PP) described in the proof of Theorem 1. Given an instance of 3PP, we construct the following instance for problem under Case (ii): $3n$ orders with the order set $N = A = \{1, \cdots, 3n\}$, each order $j \in N$ has a processing time $p_j = MH - a_j$ and a weight $w_j = MH + a_j$, the batch size is $b = 3MH + H$, the lead time threshold is $L = 1/2(3MH - H)(n+1)$, and the threshold for the number of deliveries is n, where M is a sufficiently large integer, e.g. $M \geq 3n^2 + 5n + 2$.
(If Part) If there is a partition $A_1, \cdots, A_n$ of A for the 3PP instance such that each A_i, for $i = 1, \cdots, n$, contains exactly 3 elements and its total size is H, then we construct

a schedule for the problem instance as follows. Process the orders of $A_1, \cdots, A_n$ in this sequence with an arbitrary sequence among the three orders from the same subset. For each $i = 1, \cdots, n$, the three orders of A_i complete processing at time $(3MH - H)i$, and the total size of these three orders is $3MH + H = b$. Deliver the three orders of A_i in one batch at time $(3MH - H)i$, for $i = 1, \cdots, n$. It can be seen that in this joint production and delivery schedule, the average delivery time of all the orders is exactly L and the total number of deliveries is n.

(Only If Part) Given a schedule π for the problem instance with an average delivery time no more than L and the number of deliveries no more than n, we can conclude that in schedule π there are exactly n delivery batches and each batch is full. This is because the total size of all the orders together is $3nMH + nH = nb$ and hence at least n deliveries are needed. Let V_j be the set of orders that are entirely delivered in the jth batch or whose very last parts are delivered in the jth batch (i.e. D_i is equal to the departure time of this batch for all $i \in V_j$) in schedule π. Let x_j be the number of orders in V_j, and T_j be the departure time of delivery batch j in schedule π. The following results hold in schedule π.

Claim 1: $x_1 + \cdots + x_i \leq 3i$, for $i = 1, \cdots, n$. We prove this by contradiction. Suppose for some i, $x_1 + \cdots + x_i \geq 3i + 1$, then the total weight of the orders in $V_1 \cup \cdots \cup V_i$ is more than $(3i+1)MH > i(3MH + H) = ib$. This means that more than i batches are needed to deliver those orders, which is a contradiction with the fact that those orders are delivered in exactly i batches in π.

Claim 2: $\frac{M-1}{M+1}bi \leq T_i \leq bi$, for $i = 1, \cdots, n$. By the time T_i, i full batches of orders have completed processing. To fill one unit of weight, we need at least $\min\{p_j/w_j \mid j \in N\} = (MH - a_{\max})/(MH + a_{\max}) \geq (MH - H)/(MH + H) = (M-1)/(M+1)$ units of processing time, where $a_{\max} = \max\{a_j \mid j \in A\}$. Similarly, to fill one unit of weight, we need at most

$\max\{p_j/w_j \mid j \in N\} = (MH - a_{\min})/(MH + a_{\min}) \geq 1$ unit of processing time. Thus, the time when i batches are fully filled, T_i satisfies: $\frac{M-1}{M+1} bi \leq T_i \leq bi$.

By Claim 1, and the fact that $x_1 + \cdots + x_n = 3n$, we can see that $x_i + \cdots + x_n \geq 3(n-i+1)$ for $i = 1, \cdots, n$. By Claim 2, we can see that, for $i = 1, \cdots, n-1$,

$$T_{i+1} \geq \frac{M-1}{M+1} b(i+1) = bi + b - \frac{2(bi+b)}{M+1} \geq T_i + b[1 - \frac{2n+2}{M+1}] \tag{5.7}$$

Next we show that $x_1 = \cdots = x_n = 3$ by contradiction. Suppose that there exists some $1 \leq k \leq n$ such that $x_{k+1} = \cdots = x_n = 3$ and $x_k > 3$. Define $y_{k-1} = x_{k-1} + x_k - 3$, and $y_i = x_i$ for $i = 1, \cdots, k-2$. It can be easily verified that $y_1, \cdots, y_{k-1}$ satisfy: $\sum_{j=i}^{k-1} y_j \geq 3[(k-1) - i + 1]$ for all $i = 1, \cdots, k-1$, and $\sum_{j=1}^{k-1} y_j = 3(k-1)$. Thus by Lemma 20, we have

$$\sum_{j=1}^{k-1} y_j T_j \geq 3 \sum_{j=1}^{k-1} T_j \tag{5.8}$$

Then the total delivery time of the orders in schedule π is

$$\begin{aligned}
\sum_{i=1}^{n} x_i T_i &= \sum_{i=k+1}^{n} 3T_i + x_k T_k + \sum_{i=1}^{k-1} x_i T_i \\
&> \sum_{i=k+1}^{n} 3T_i + 3T_k + (x_k - 3)(T_{k-1} + b[1 - \frac{2n+2}{M+1}]) + \sum_{i=1}^{k-1} x_i T_i \quad \text{(by (5.7)} \\
&> \sum_{i=k}^{n} 3T_i + b[1 - \frac{2n+2}{M+1}] + (x_{k-1} + x_k - 3) T_{k-1} + \sum_{i=1}^{k-2} x_i T_i \quad \text{(since } x_k > 3) \\
&= \sum_{i=k}^{n} 3T_i + b[1 - \frac{2n+2}{M+1}] + \sum_{j=1}^{k-1} y_j T_j \\
&\geq \sum_{i=k}^{n} 3T_i + b[1 - \frac{2n+2}{M+1}] + 3 \sum_{j=1}^{k-1} T_j \quad \text{(by (5.8))} \\
&= \sum_{i=1}^{n} 3T_i + b[1 - \frac{2n+2}{M+1}] \\
&\geq \frac{3b(n^2+n)(M-1)}{2(M+1)} + b[1 - \frac{2n+2}{M+1}] \quad \text{(by Claim 2)} \\
&= 3/2b(n^2+n) + \frac{b}{M+1}(M - 3n^2 - 5n - 1) \\
&> 3nL, \quad \text{(by the fact that } M > 3n^2 + 5n + 1 \text{ and } b > 3MH - H)
\end{aligned}$$

which is in contradiction with the fact that the mean delivery time of the orders in π is no more than L. Therefore, $x_1 = \cdots = x_n = 3$. This means that in schedule π, for every $i = 1, \cdots, n$, the first i delivery batches together deliver the first $3i$ orders and possibly a part of the $(3i+1)$th order. Let the processing time and weight of the part of the $(3i+1)$th order covered in the ith batch be denoted as α_i and β_i. If the ith batch does not cover a part of the $(3i+1)$th order, we can simply let $\alpha_i = \beta_i = 0$. Let the processing sequence of orders under schedule π be denoted as $([1], \cdots, [3n])$. The total weight of the orders covered in the first i batches is $3iMH + \sum_{j=1}^{3i} a_{[j]} + \beta_i = ib$, which implies that

$$\sum_{j=1}^{3i} a_{[j]} = iH - \beta_i \tag{5.9}$$

The total processing time of the orders covered in the first i batches is $T_i = 3iMH - \sum_{j=1}^{3i} a_{[j]} + \alpha_i$. By (5.9), we have $T_i = 3iMH - iH + \alpha_i + \beta_i$. Since the average delivery time of the orders in π, $\frac{3\sum_{i=1}^{n} T_i}{3n}$ must be no more than L, we have $\alpha_i = \beta_i = 0$ for all $i = 1, \cdots, n$. This, together with (5.9), means that $\sum_{j=1}^{3i} a_{[j]} = iH$ for $i = 1, \cdots, n$. Therefore, $\sum_{j=3i-2}^{3i} a_{[j]} = H$ for $i = 1, \cdots, n$, and the subsets $\{[1], [2], [3]\}, \{[4], [5], [6]\}, \cdots, \{[3n-2], [3n-1], [3n]\}$ form a solution to the 3PP instance. ■

Theorem 23 The lead time problem with splittable production and delivery is strongly NP-hard.

Proof It can be easily checked that all the results proved in the proof of Theorem 22 up to the result $x_1 = \cdots = x_n = 3$ in the "Only If" part apply to the problem with Case (iii) as well. The arguments given there after the result $x_1 = \cdots = x_n = 3$ is shown are applicable to Case (ii) only, but can be slightly modified as follows to work for the problem with Case (iii). The result $x_1 = \cdots = x_n = 3$ means that in schedule π, for every $i = 1, \cdots, n$, the first i delivery batches together deliver $3i$ orders and possibly parts of some other orders. Let the total processing time and total weight of the parts of the other orders covered in these batches be denoted as α_i and β_i. Let the completion sequence of orders under schedule π be denoted as $([1], \cdots, [3n])$. The rest of the proof is the exactly the same as that in the proof of Theorem 22. ■

5.4.2 Heuristics for Cases (i), (ii) and (iii)

We first propose a heuristic for Case (i) of the lead time problem. The heuristic is a dynamic programming based approach. For any given number of delivery batches m, the dynamic program involved finds a schedule with a minimum total delivery time among a subset of feasible schedules with exactly m delivery batches. The dynamic program builds up a schedule step by step from time 0 onward, and in each step a subset of orders is scheduled for processing and delivery. To schedule a given subset of orders $Q \subseteq N$ starting from a given time t in a particular step, the following procedure is used.

Procedure BFD

Input: A subset of orders Q and a starting time t for the first order.

Step 1: Assign the orders of Q to delivery batches using the well-known Best-Fit-Decreasing (BFD) rule (see, e.g. Coffman et al. 1997) for the classical bin-packing problem. Let h be the number of batches formed. Let P_k be the total processing time of the orders assigned to batch k, for $k = 1, \cdots, h$.

Step 2: Let $\tau = t$. For $k = 1, \cdots, h$, process the orders of batch k from time τ to $\tau + P_k$ and deliver this batch at time $\tau + P_k$, and update $\tau = \tau + P_k$. The total delivery time of the orders of Q can be calculated accordingly.

Next we describe the heuristic. Let $m = \lceil \sum_{j \in N} w_j / B \rceil$. Clearly, at least m delivery batches is necessary to deliver the n orders of N.

Heuristic H3

Step 0: Reindex the orders of N such that $p_1 \leq \cdots \leq p_n$.

Step 1: Run the following dynamic programming algorithm.

Define value function $F(i, j)$ to be the minimum total delivery time of the first j orders $\{1, \cdots, j\}$ given that they are processed from time 0 without idle time and that they are delivered in i batches.

Initial conditions: $F(0, 0) = 0$ and $F(i, j) = \infty$ for any (i, j) satisfying: $i < 0$ or $i = 0$ and $j > 0$. Recursive relations: For $i = m, \cdots, n$, and $j = 1, \cdots, n$,

$$F(i, j) = \min\{F(i - g(v + 1, j), v) + G(v + 1, j) \mid v = 0, \cdots, j - 1\},$$

where $g(v + 1, j)$ is the number of delivery batches formed by applying the procedure BFD to the subset of orders $Q = \{v+1, \cdots, j\}$ with the starting time $t = \sum_{j=1}^{v} p_j$, and $G(v+1, j)$ is the corresponding total delivery time of the orders of Q.

Solutions: For $i = m, \cdots, n$, $F(i, n)$ is the minimum total delivery time of all the n orders among all the schedules with exactly i delivery batches that are covered by the dynamic program. Let the corresponding schedule be $S(i, n)$.

Step 2: Let q be the minimum possible i ($m \leq i \leq n$) with $F(i, n) \leq nL$. The schedule found by this heuristic is $S(q, n)$ and the number of delivery batches is q.

The dynamic program considers all the schedules with the following structure: The order sequence can be divided into a number of blocks such that (i) orders across different blocks are scheduled in SPT order, (ii) the orders within a block are scheduled by the procedure BFD and consequently they are divided into one or more subsets by the BFD rule, each delivered by a separate batch. Note that the orders within each block is not necessarily scheduled in SPT order. For ease of presentation, we call a schedule with the above structure a block-SPT-BFD schedule. The schedule found by the heuristic $S(q, n)$ is optimal among all the block-SPT-BFD schedules.

Theorem 24 The worst-case performance ratio of Heuristic H3 for the lead time problem with non-splittable production and delivery is bounded by 3.

Proof Given an optimal schedule π for the problem under Case (i), let z^* and L^* be the number of delivery batches used and the mean delivery time of the orders in π, respectively. Clearly, $L^* \leq L$. Let n_i^* denote the number of orders and T_i^* the completion time of the last order in the ith batch of π, for $i = 1, \cdots, z^*$. We construct a block-SPT-BFD schedule ρ based on π using the following procedure:

(i) Sequence the n orders in SPT order. Without loss of generality, suppose this sequence is $(1, \cdots, n)$.

(ii) Divide this sequence into z^* blocks of consecutive orders, denoted as $R_1, \cdots, R_{z^*}$, such that the ith block R_i consists of the n_i^* orders: $\sum_{u=1}^{i-1} n_u^* + 1, \cdots, \sum_{u=1}^{i} n_u^*$. Let the completion time of the last order of R_i be E_i. It can be easily shown that

$$E_i \leq T_i^*, \quad \text{for } i = 1, \cdots, z^*. \tag{5.10}$$

(iii) For $i = 1, \cdots, z^*$, if the total weight of the orders in R_i, denoted as W_i, is no more than the batch size b, then deliver all the orders of R_i in a single batch at time E_i. Otherwise, apply the procedure BFD to the orders $Q = R_i$ with starting time $t = E_{i-1}$. Let r_i be the resulting number of delivery batches covering the orders of R_i. Clearly, the delivery time of each batch is no more than E_i. By (5.10), we can conclude that the average delivery time of the orders in ρ is no more than that in π and hence is no more than L. This means that ρ is a feasible schedule.

In the above procedure (iii), for $i = 1, \cdots, z^*$, if $W_i > b$, then $W_i > r_i b/2$ for the following reason. For $k = 1, \cdots, r_i$, let X_k be the total weight of the orders assigned to the kth delivery batch. Thus $W_i = \sum_{k=1}^{r_i} X_k$. By the BFD rule used to assign orders to batches, we can see that $X_u + X_{u+1} > b$ for $u = 1, \cdots, r_i - 1$ and $X_{r_i} + X_1 > b$. Summing both sides of these r_i inequalities up, we have $\sum_{u=1}^{r_i} X_u \geq r_i b/2$. Define set $H_1 = \{i \mid r_i = 1, 1 \leq i \leq z^*\}$ and $H_2 = \{1, \cdots, z^*\} \setminus H_1$. Denote the total number of deliveries in schedule ρ by $q(\rho)$. Then,

$$
\begin{aligned}
q(\rho) &= |H_1| + \sum_{i \in H_2} r_i \\
&\leq z^* + (2 \sum_{i \in H_2} W_i)/b \\
&\leq z^* + 2(\sum_{i=1}^{z^*} W_i)/b \\
&\leq z^* + 2z^* \\
&= 3z^* \qquad (5.11)
\end{aligned}
$$

Since the dynamic program in the heuristic considers all block-SPT-BFD schedules including ρ, the schedule found by the heuristic $S(q, n)$ has the number of delivery batches q no more than $q(\rho)$. This, together with (5.11), shows that the number of delivery batches in schedule $S(q, n)$ is at most 3 times that the optimal number of delivery batches. ■

Next, we propose a heuristic for Cases (ii) and (iii) of the lead time problem. The general idea and structure of the heuristic are similar to that of the heuristic H3 for Case (i) of the problem. It is also dynamic programming based and the DP tries to find optimal schedules among a subset of feasible schedules only. However, since partial delivery of an order is allowed in Cases (ii) and (iii) of the problem, the procedure used to schedule a given subset of orders in each step of the DP is different. As in the heuristic H3, the dynamic program involved builds up a schedule step by step from time 0 onward. In each step, a subset of consecutive orders is taken from the initial SPT sequence of the orders and scheduled for processing and delivery by the following procedure FB with Q being the subset of the orders to be considered and t the starting time for the processing of the first order of Q.

Procedure FB

Input: A subset of orders Q and a starting time for the first order t.

Step 1: Specify a sequence of the orders in Q and denote it by $([1], \cdots, [u])$, where $u = |Q|$.

Step 2: Let $h = \lceil \sum_{j=1}^{u} w_{[j]}/b \rceil$. Assign the orders of Q to h delivery batches using the following full-batch (FB) rule. Take the whole orders $[1], \cdots, [i_1]$ and a portion α $(0 < \alpha < 1)$ of order $[i_1+1]$ such that $\sum_{j=1}^{i_1} w_{[j]} + \alpha w_{[i_1+1]} = b$, and assign them to the first delivery batch. Take the remaining part of order $[i_1+1]$ and a number of whole orders $[i_1+2], \cdots, [i_2]$ and possibly a partial order $[i_2+1]$ such that their total weight is exactly b, and assign them to the second delivery batch. Repeat the above until all the orders are assigned. Let h be the number of batches formed. Clearly, all the batches except possibly the last one are full. Let P_k be the total processing time of the orders assigned to batch k, for $k = 1, \cdots, h$.

Step 3: Let $\tau = t$. For $k = 1, \cdots, h$, process the orders of batch k from time τ to $\tau + P_k$ and deliver this batch at time $\tau + P_k$, and update $\tau = \tau + P_k$. The total delivery time of the orders of Q can be calculated accordingly.

Our heuristic H4 for Case (ii) (and Case (iii)) is exactly the same as the heuristic H3 except that within the DP algorithm the procedure FB is used to schedule a subset of orders Q. We omit the details of the heuristic.

Heuristic H4 considers all the schedules with the following structure: The order sequence can be divided into a number of blocks such that (i) orders across different blocks are scheduled in SPT order, (ii) the orders within a block are scheduled by the procedure FB and consequently they are divided into one or more subsets by the FB rule, each delivered by a separate batch. We call a schedule with the above structure block-SPT-FB schedule. If we always use the same relative sequence of orders in Step 1 of the procedure FB, then the schedule found by the heuristic $S(q, n)$ is optimal among all the block-SPT-FB schedules where the orders in each block follow the same relative sequence.

Theorem 25 The worst-case performance ratio of Heuristic H4 for Cases (ii) and (iii) of the lead time problem is bounded by 2.

Proof Given an optimal schedule π for Case (ii) or (iii) of the problem, let z^* and L^* be the number of delivery batches used and the mean delivery time of the orders in π, respectively. Clearly, $L^* \leq L$. Let T_i be the delivery time of the ith batch in π. For $i = 1, \cdots, z^*$, let n_i^* denote the number of orders who are either entirely covered or whose last part is covered in the ith batch of π.

We construct a block-SPT-FB schedule ρ based on π by the following procedures:

(i) Sequence the n orders in SPT order. Without loss of generality, suppose this sequence is $(1, \cdots, n)$.

(ii) Divide this sequence into z^* blocks of consecutive orders, denoted as $R_1, \cdots, R_{z^*}$, such that the ith block R_i consists of the n_i^* orders: $\sum_{u=1}^{i-1} n_u^* + 1, \cdots, \sum_{u=1}^{i} n_u^*$.

(iii) For $i = 1, \cdots, z^*$, apply the procedure FB (where the same sequence of orders as the one used in the heuristic is used in Step 1) to the orders $Q = R_i$ with starting time $t = E_{i-1}$ (where $E_0 = 0$). Let r_i be the resulting number of delivery batches covering the orders of R_i. By a similar argument as in the proof of Theorem 24, it can be shown that the departure time of each of the r_i batches corresponding to block R_i is no later than T_i, and hence ρ is a feasible block-SPT-FB schedule.

In the above procedure (iii), since the procedure FB generates for each block at most one batch which is less than full, the total weight of the orders in R_i, denoted as W_i, satisfies,

$$W_i \geq (r_i - 1)B, \quad \text{for } i = 1, \cdots, z^* \tag{5.12}$$

Denote the total number of deliveries in schedule ρ by $q(\rho)$. Then, by (5.12), we have

$$\begin{aligned} q(\rho) &= \sum_{i=1}^{z^*} r_i = z^* + \sum_{i=1}^{z^*} (r_i - 1) \\ &\leq z^* + (\sum_{i=1}^{z^*} W_i)/B \\ &\leq z^* + z^* = 2z^* \end{aligned} \tag{5.13}$$

Since the dynamic program in the heuristic considers all block-SPT-FB schedules including ρ, the schedule $S(q, n)$ found by the heuristic has the number of delivery batches q no more than $q(\rho)$. This, together with (5.13), shows that the number of delivery batches in schedule $S(q, n)$ is at most 2 times that the optimal number of delivery batches. ■

5.4.3 Computational Experiment

We use column generation to obtain a lower bound for each case of the lead time problem. These lower bounds are used to evaluate the performance of Heuristics H3 and H4 proposed in the previous subsection.

Column generation for Case (i)

In this section, we describe the column generation based approach to obtain a lower bound for Case (i). Since the approach is very similar to the one used for the deadline problem under Case (i) in Section 5.3.3, we only give a brief description.

It can be easily proved that the number of delivery batches in an optimal solution for Case (i) is a non-increasing function of the threshold value L on the mean lead time. We use this property to get a lower bound of Case (i). We first consider the following dual problem: Find the minimum mean lead time of the orders subject to the constraint that no more than m delivery batches can be used. Let $Z_{dual}(m)$ be the optimal objective value of this dual problem. Then $m^* = \min\{m \mid Z_{dual}(m) \leq L\}$ is the optimal objective value of Case (i). If $LB_{dual}(m)$ is a lower bound of $Z_{dual}(m)$, then it can be easily shown that $m' = \min\{m \mid LB_{dual}(m) \leq L\}$ is a lower bound of the optimal objective value of Case (i). Based on this observation, we can find a lower bound of the optimal objective value of Case (i) as follows.

We first formulate the dual problem as the following integer program, where the parameters $\Omega, \omega, a_{i\omega}, b_{i\omega}$ are all defined exactly the same way as in Section 5.3.3:

$$[\text{IP2}] \quad \min \frac{1}{n} \sum_{\omega \in \Omega} f_\omega x_\omega \tag{5.14}$$

Subject to:

$$\sum_{\omega \in \Omega} x_\omega \leq m \tag{5.15}$$

$$\sum_{\omega \in \Omega} a_{j\omega} x_\omega = 1 \qquad \forall j \in N \tag{5.16}$$

$$\sum_{\omega \in \Omega} b_{t\omega} x_\omega = 1 \qquad \forall t \in \{0, \cdots, P-1\} \tag{5.17}$$

$$x_\omega \in \{0, 1\} \qquad \forall \omega \in \Omega \tag{5.18}$$

In [IP2], f_ω is the sum of delivery times for the orders in ω. Constraint (5.15) ensures that there are no more than m delivery batches. The rest of the constraints are the same as the ones in [IP1]. We use [LP2] to denote the LP relaxation of [IP2] and $Z^*_{LP2(m)}$ to denote the optimal objective value of [LP2] when the number of batches that can be used is restricted to m. Our algorithm for obtaining a lower bound for Case (i) using [LP2] is described next:

Algorithm A2

Step 0: Run heuristic H3 and let the resulting solution be Z^{H3}. Set $k^* = Z^{H3}$ and $k = Z^{H3} - 1$.

Step 1: Solve [LP2] with $m = k$.

Step 2.1: If $Z_{LP2(k)} \leq L$, and $k > \lceil \sum_{j \in N} w_j / b \rceil$, set $k^* = k$ and $k = k - 1$. Goto Step 1.

Step 2.2: If $Z_{LP2(k)} \leq L$, and $k = \lceil \sum_{j \in N} w_j / b \rceil$, STOP. $k^* = k$ is a valid lower bound for Case (i).

Step 2.3: If $Z_{LP2(k)} > L$, or $k = \lceil \sum_{j \in N} w_j / b \rceil$, STOP. k^* is a valid lower bound for Case (i).

Algorithm A2 decreases the value of m till the optimal objective value of [LP2] goes above the allowed threshold L, or till the number of allowed batches reaches its lower bound. Since $Z^{H3} \leq n$, the number of iterations involved in Algorithm A2 is at most n.

We use a column generation approach to solve [LP2] at every iteration. The columns are added using a dynamic programming approach which is very similar to the one described in Section 5.3.3. Without loss of generality, we assume that the orders are indexed in the SPT order, with ties broken arbitrarily. To initialize the column generation procedure, a dummy column with a very high objective value is introduced. We use γ, π_j and σ_t to denote the dual variable value corresponding to the constraint sets (5.15), (5.16) and (5.17) respectively. Then the reduced cost r_ω of a column corresponding to $\omega \in \Omega$ is given by: $r_\omega = f_\omega / n - \gamma - \sum_{j \in \omega} \pi_j - \sum_{t \in [t_{\omega s}, t_{\omega e} - 1]} \sigma_t$.

In the following, we describe the dynamic programming (DP) algorithm for solving the subproblem.

Algorithm DP2

Define the value function $F_e(j,t,w)$ as the minimum reduced cost of a batch consisting of a subset of the first j orders $\{1,\cdots,j\}$, given that the batch departs at time e, the total current weight of the batch is w and the total processing time of the orders in the batch is t, where $0 \le e \le P$, $1 \le j \le n$, $0 \le t \le e$, $0 \le w \le b$.

Initial Values

$F_e(0,0,0) = -\gamma$ for $0 \le e \le P$.

$F_e(j,t,w) = \infty$ for any state (e,j,t,w) violating any of the following conditions: $0 \le e \le P$, $1 \le j \le n$, $0 \le t \le e$, $0 \le w \le b$.

Recursive Relations

$$F_e(j,t,w) = \min\left\{F_e(j-1,t,w),\quad F_e(j-1,t-p_j,w-w_j) + e/n - \pi_j - \textstyle\sum_{\tau\in[e-t,e-(t-p_j+1)]}\sigma_\tau\right\}$$

Optimal Solution is obtained by solving:

$\min\{F_e(j,t,w) | 1 \le j \le n, p_{\min} \le e \le P, p_{\min} \le t \le e, 0 \le w \le b\}$, where $p_{\min}$ is the minimum processing time among all the orders.

Similar to Algorithm DP1, we can show that DP2 solves the problem of finding the minimum reduced cost optimally in $O(P^2nb)$ time.

Column generation for Cases (ii) and (iii)

In this section, we use column generation to obtain a lower bound for Cases (ii) and (iii). Since the approach is very similar to the one used for Case (i), we only give a brief description.

Similar to Case (i) of the problem, we do a search on the number of batches. But instead of using an exact formulation, we formulate a relaxed problem. This is done because in the exact formulation, the complexity of the subproblem is much more than that for Case (i) and hence it can be extremely time-consuming to obtain the optimal solutions of the LP relaxations even for small sized problems. In the relaxed problem, we assume that production is splittable for both cases and for each order delivered in multiple batches, a lower bound is used for the lead time of the order. For any order j, if α_j units of its weight w_j is included in a batch with departure time e, we measure the contribution to the lead time of order j made by this partial order as $(\alpha_j/w_j)e$. By lemma 18, we need to check for only integer values of α_j. In order to ensure that all the DP states have integer values for t, in the test problems we set the processing time of an order j as $\tau_j w_j$, where τ_j is a positive integer drawn randomly. This ensures that we have a finite number of states for the dynamic program. The master LP formulation, denoted as [LP3] is exactly the same as [LP2], except that in each batch $\omega \in \Omega$, we now allow any number of partial orders to be included as long as their total weight does not exceed the batch capacity.

In the DP for the subproblem (denoted as DP3), we define value function $F_e(j, t, w)$ similar to Case (i). At each stage, the recursive relations check for every possible value of α_j for an order j. Since the details of the DP are very similar to that of DP2, we only give the recursive relations:

Recursive Relations

$$F_e(j,t,w) = \min\{F_e(j-1,t,w), \min_{\alpha_j \in [1,w_j]}\{F_e(j-1, t-\alpha_j p_j/w_j, w-\alpha_j)+ \frac{e\alpha_j}{w_j n} - \pi_j \alpha_j/w_j - \textstyle\sum_{\tau \in [e-t, e-t+\alpha_j p_j/w_j - 1]} \sigma_\tau\}\}$$

Since $w_j \leq b$ for any order j, the computational complexity of the recursive relations is given by $O(b)$. Proceeding along the same lines as lemma 19, we can show that DP3 solves the problem of finding the minimum reduced cost optimally in $O(P^2nb^2)$ time.

Since any feasible solution to Case (ii) and Case (iii) is also feasible to [LP3], and since the contribution by $\omega \in \Omega$ in [LP3] to the objective function is a lower bound on the actual contribution in Case (ii) and Case (iii), [LP3] gives a valid lower bound to both the problems.

Computational results

In this section, we compare heuristics H3 and H4 with valid lower bounds obtained through column generation. The parameter settings for H3 are exactly the same as the ones used in Section 5.3.3 except that instead of the deadlines, we vary the lead time threshold to obtain a wide range for the number of delivery batches. For H4, the settings are similar to that of H3. The only differences are in the number of orders and the way the processing times for the orders are derived. Since the column generation approach is more computationally involved for [LP3], we could test only problems with 20, 25, and 30 orders. The processing times for the orders are set as $x_j w_j$, where x_j is selected randomly from the set $\{1, 2, 3\}$ and w_j is the weight of order j (this is to ensure that the DP is tractable, as mentioned in the description of the column generation approach).

Table 5.3 reports the results for Case (i). The overall average and maximum gaps are given by 1.64% and 11.11% respectively. The average gaps for the High, Medium, and Low sum of lead time cases are 0.78%, 1.12%, and 3.03% respectively. The heuristic performs better under tight lead times because when the lead time constraint is tight, most orders get shipped independently and hence most of the delivery batches are not full even in the optimal solution. When the lead time constraint is loose, delivery batches consolidate multiple orders and the performance of the heuristic is bound by the best-fit decreasing heuristic that is used to batch the orders in the dynamic program. The average gaps for the 20, 40, and 60 order instances are 1.43%, 1.97%, and 1.53% respectively. Similarly, the average gaps for the three different average order sizes are: 0.53%, 2.27%, and 2.13% respectively. Here again, we do not observe any clear trend.

Table 5.4 reports the results for H4. The overall average and maximum gaps are given by 5.01% and 30.00% respectively. We note that the gaps may be high due to the fact that the column generation formulation is for a relaxed version of the problem. In the column generation formulation, the lead time for an order is counted as the weighted sum of the lead times of the fractions of the order that get delivered. Besides, the heuristic does not preempt the orders during production in accordance with Case (ii), whereas the column generation formulation allows for order preemption. The average gaps for the High, Medium, and Low sum of lead time cases are 3.37%, 4.05%, and 7.62% respectively. Once again, the heuristic performs better under tight lead times because when the lead time constraint is tight, most orders get shipped independently and hence most of the delivery batches are not full even in the optimal solution. When the lead time constraint is loose, delivery batches consolidate multiple orders and the performance of the heuristic is not as good. The average gaps for the 20, 25, and 30 order instances are 4.15%, 5.19%, and 5.69% respectively. This increase may be because the lower bound is not as tight with a higher number of orders. With a higher number of orders, an order could be processed as part of two batches that are quite far away from each other, and when this happens, the weighted sum of their lead times is going to be much lower than the actual lead time. The average gaps for the three different average order sizes are: 2.39%, 7.35%, and 5.28% respectively. Here again, we do not observe any clear trend.

5.5 Extensions

In this section, we consider two extensions of the model considered in the earlier sections. The first extension considers inventory cost incurred when a completed order has to wait for some other orders to complete so that they can be delivered together in the same batch. The second extension considers the case when some of the orders are not known at time zero and instead they arrive randomly over time. We look at both extensions in the context of the deadline problem under Case (i).

5.5.1 Inventory Consideration

We define some additional notation. In a given schedule, we define the waiting time of order $j \in N$ to be $I_j = D_j - C_j$, where C_j and D_j are defined in Section 5.2 to be the completion time of order j and the departure time of the batch containing order j, respectively. Let h be the unit inventory cost per period and f the delivery cost per batch. The objective of the problem is to minimize the total inventory and delivery cost, that is, $h \sum_{j \in N} I_j + fx$, where x is the number of delivery batches used, subject to the constraint that all the orders are delivered to the customer no later than their deadlines, i.e. $D_j \leq d_j$ for $j \in N$. The problem is clearly strongly NP-hard as it is more general than Case (i) of the deadline problem considered in Section 5.3.

It is easy to see that all the orders delivered in the same batch should be processed in LPT order because LPT order minimizes the inventory cost. We generalize heuristic H1 to this problem as follows. First generate a solution by this heuristic. Then reschedule the orders for processing by the following rule: Orders delivered earlier are scheduled earlier, and orders within the same batch are scheduled in LPT order.

Next we evaluate the performance of this heuristic by comparing the heuristic results with valid lower bounds generated by a column generation approach. The column generation approach is similar to the one provided for Case (i) of the deadline problem except that in the subproblem, the orders are initially sequenced in SPT order so that the resulting processing sequence for the orders in the same batch becomes LPT, and the inventory costs are taken into account while calculating the reduced costs.

We tried three different values of per period inventory holding cost h, $h \in \{0.001, 0.01, 0.1\}$, for problem instances with 40 orders. For all the test instances, f, the delivery cost per batch, was set as 1. Table 5.5 gives the results of the computational experiment. We note that the average and maximum gaps when there is no inventory cost (given in Table 5.1) are 6.17% and 16.67% respectively with 40 orders. We see an increase in the gaps as soon as an inventory cost is introduced. This is expected since the heuristic does not take into account the inventory costs while forming the batches, though it does try to improve the batches formed by processing all the orders in a batch consecutively in the LPT order. The average gap with $h = 0.001, 0.01$, and 0.1 are given by 7.94%, 7.25% and 18.50% respectively. Similarly, the maximum gap with $h = 0.001, 0.01$, and 0.1 are given by 20.00%, 16.49% and 71.01% respectively. We see that the gap decreases when h is changed from 0.001 to 0.01. This is because though the total costs in both the heuristic and the column generation outputs increase, the cost function is still dominated by the delivery costs and hence the ratio of the two drops. On the other hand, when $h = 0.1$, inventory costs start taking over and since the heuristic is not very efficient in considering the inventory costs, the gaps increase.

5.5.2 Random Order Arrivals

Let N_1 be the set of n_1 orders that have arrived at time zero. In a given period of time, say, $[0, T]$, where T may or may not be greater than $\sum_{j \in N_1} p_j$, a small set of n_2 orders N_2 may arrive randomly over time. We do not have any knowledge about the distribution of any parameter associated with those orders. We assume that each order that arrives after time zero has a sufficiently large deadline such that we can always meet its deadline if it is processed immediately after the first order currently being processed in the system. The objective of the problem is to minimize the number of delivery batches subject to the constraint that all the orders including both N_1 and N_2 are delivered to the customer no later than their deadlines, i.e. $D_j \leq d_j$ for $j \in N_1 \cup N_2$. The problem is clearly strongly NP-hard as it is more general than Case (i) of the deadline problem considered in Section 5.3.

We generalize heuristic H1 to this problem as follows. Run the heuristic for the current orders in the system. Whenever a new order arrives, first process it by inserting it to a position so that EDD order sequence is maintained among all the existing orders. Then continue the heuristic.

Next we evaluate the performance of this heuristic by comparing its results to global lower bounds generated by a column generation approach. The column generation approach assumes that all the $n_1 + n_2$ orders are known in advance and are available at time zero.

We tested problem instances with a total of 40 orders where a fraction x of orders arrived randomly over time. The results are given in Table 5.6. We tried for three different values of x: $x \in \{0.1, 0.2, 0.3\}$. All the other parameter settings are exactly the same as the ones used in Section 5.3.3. The average gaps were 6.30%, 6.39%, and 6.19% for $x = 0.1$, $x = 0.2$, and $x = 0.3$ respectively. The maximum gap was 16.67% for all the three cases. We note that the average and the maximum gaps were 6.17% and 16.17% respectively for the 40 order case in Section 5.3.3. Thus there is only a very marginal increase in the average gap. This can be explained as follows: When the deadlines are tight, most of the orders anyway get processed close to their deadlines and as long as an order arrives with sufficient time left for its processing, the production schedule will not change in general. On the other hand, when the deadlines are not tight, the production sequence does not make much of a difference to the heuristic since most of the orders will anyway be able to combine with most of the other orders, whichever sequence they are processed in. Hence, though the production sequence may get affected, there may not be any effect on the final number of delivery batches. So as long as the fraction of orders that arrive at time $t > 0$ is not very large, or as long as the late orders are well spread over the time horizon, the effect on the final solution may not be significant. But when x approaches 1, or when all the late orders arrive at a time very close to their deadlines, other complications may arise. One example is the non-availability of any order for processing at a particular time because none of the remaining orders have arrived. This may lead to idle time in the schedule and may even lead to an infeasible problem instance.

5.6 Conclusions

In this study, we have analyzed an integrated production-distribution scheduling model in a supply chain with one supplier and one customer. We have considered a scenario where the orders generally have different sizes while the delivery batch capacity is finite. Production-distribution scheduling decisions have to be made jointly with the order packing decisions. Our objective was to minimize the distribution costs while ensuring that a time related service constraint is satisfied. Computational complexity of various cases of the problem have been clarified and we have provided either polynomial time optimal algorithms or fast and efficient heuristics for all the cases. We also looked at an extension that considered inventory costs and another extension that allowed for changes in the schedule to accommodate random arrivals.

Some of the production-distribution scheduling models in the literature look at settings with multiple suppliers and/or multiple customers. But in those problems, order packing is not considered. In our model, order sizes add one more dimension to the problem making it more complicated. Still it will be interesting to look at such setups and there may be special cases of the problems with multiple suppliers and/or multiple customers that are solvable in polynomial time. But we conjecture that the general versions of all those problems will be strongly NP-hard. Another extension to the model includes making delivery costs dependent on the weight carried by each batch (eg. having a fixed cost per batch and a variable cost per capacity utilized). Again, since our problem is a special case of this version, all the NP-hard problems of our model will still be NP-hard for this extension.

Table 5.1: Average gap for the deadline problem under Case (i)

Number of orders	Maximum Order Weight	Due Date Tightness	Average Gap	Maximum Gap
20	10	High	0.00%	0.00%
		Medium	6.69%	16.67%
		Low	6.87%	16.67%
	15	High	2.44%	6.67%
		Medium	7.41%	22.22%
		Low	6.76%	11.11%
	20	High	2.48%	7.14%
		Medium	5.78%	16.67%
		Low	7.19%	8.33%
40	10	High	1.89%	5.88%
		Medium	5.36%	16.67%
		Low	8.77%	16.67%
	15	High	1.90%	9.52%
		Medium	6.09%	11.11%
		Low	11.02%	11.76%
	20	High	2.13%	7.69%
		Medium	7.45%	13.04%
		Low	10.87%	13.64%
60	10	High	1.54%	7.69%
		Medium	5.61%	13.33%
		Low	6.78%	14.29%
	15	High	3.03%	9.68%
		Medium	4.86%	14.29%
		Low	5.94%	9.52%
	20	High	1.46%	5.56%
		Medium	5.12%	14.81%
		Low	6.79%	14.81%
		Overall	5.27%	22.22%

Table 5.2: Average gap for the deadline problem under Case (ii)

Number of orders	Maximum Order Weight	Due Date Tightness	Average Gap	Maximum Gap
20	10	High	0.00%	0.00%
		Medium	0.00%	0.00%
		Low	2.50%	12.50%
	15	High	0.00%	0.00%
		Medium	0.00%	0.00%
		Low	1.43%	7.14%
	20	High	0.00%	0.00%
		Medium	1.33%	6.67%
		Low	0.00%	0.00%
40	10	High	0.00%	0.00%
		Medium	0.00%	0.00%
		Low	1.11%	5.56%
	15	High	0.00%	0.00%
		Medium	0.69%	3.45%
		Low	1.05%	5.26%
	20	High	0.56%	2.78%
		Medium	2.14%	4.35%
		Low	1.74%	4.35%
60	10	High	0.00%	0.00%
		Medium	0.00%	0.00%
		Low	1.43%	7.14%
	15	High	0.57%	2.86%
		Medium	0.87%	4.35%
		Low	0.71%	3.57%
	20	High	0.00%	0.00%
		Medium	1.38%	6.90%
		Low	1.06%	2.86%
		Overall	0.69%	12.50%

Table 5.3: Average gap for the lead time problem under Case (i)

Number of orders	Maximum Order Weight	Completion Time Tightness	Average Gap	Maximum Gap
20	10	High	0.00%	0.00%
		Medium	0.00%	0.00%
		Low	0.00%	0.00%
	15	High	1.18%	5.88%
		Medium	1.43%	7.14%
		Low	4.22%	11.11%
	20	High	2.35%	5.88%
		Medium	0.00%	0.00%
		Low	3.67%	10.00%
40	10	High	0.00%	0.00%
		Medium	0.00%	0.00%
		Low	2.35%	5.88%
	15	High	1.18%	2.94%
		Medium	2.25%	3.85%
		Low	4.27%	5.56%
	20	High	1.16%	2.94%
		Medium	2.14%	3.57%
		Low	4.38%	5.00%
60	10	High	0.00%	0.00%
		Medium	1.00%	2.50%
		Low	1.43%	3.57%
	15	High	0.38%	1.92%
		Medium	1.45%	2.44%
		Low	4.05%	6.90%
	20	High	0.75%	1.89%
		Medium	1.82%	2.33%
		Low	2.90%	5.56%
		Overall	1.64%	11.11%

Table 5.4: Average gap for the lead time problem under Case (ii) and Case (iii)

Number of orders	Maximum Order Weight	Completion Time Tightness	Average Gap	Maximum Gap
20	5	High	1.18%	5.88%
		Medium	3.08%	7.69%
		Low	0.00%	0.00%
	7	High	2.35%	5.88%
		Medium	3.08%	7.69%
		Low	10.89%	11.11%
	10	High	4.93%	12.50%
		Medium	4.40%	7.69%
		Low	7.48%	10.00%
25	5	High	2.81%	4.76%
		Medium	1.25%	6.25%
		Low	4.00%	10.00%
	7	High	2.86%	4.76%
		Medium	3.60%	6.25%
		Low	19.09%	30.00%
	10	High	2.81%	4.76%
		Medium	6.03%	6.25%
		Low	4.29%	7.14%
30	5	High	4.68%	8.00%
		Medium	3.00%	5.00%
		Low	1.54%	7.69%
	7	High	3.94%	8.00%
		Medium	5.00%	10.00%
		Low	15.38%	15.38%
	10	High	4.74%	8.00%
		Medium	7.00%	10.00%
		Low	5.89%	6.25%
		Overall	5.01%	30.00%

Table 5.5: Average gap for the deadline problem under Case (i) with inventory costs

Holding Cost	Maximum Order Weight	Due Date Tightness	Average Gap	Maximum Gap
0.001	10	1	2.53%	6.22%
		1.1	7.46%	18.25%
		1.3	13.04%	20.00%
	15	1	3.13%	12.97%
		1.1	7.57%	12.14%
		1.3	12.90%	14.91%
	20	1	2.86%	9.81%
		1.1	8.86%	15.52%
		1.3	13.14%	16.45%
0.01	10	1	1.95%	4.98%
		1.1	6.02%	13.91%
		1.3	10.32%	15.17%
	15	1	2.99%	11.81%
		1.1	7.15%	12.03%
		1.3	11.88%	13.51%
	20	1	2.90%	9.88%
		1.1	8.77%	15.24%
		1.3	13.30%	16.49%
0.1	10	1	9.16%	30.96%
		1.1	30.74%	49.68%
		1.3	48.12%	71.01%
	15	1	6.61%	18.27%
		1.1	14.87%	20.78%
		1.3	24.63%	52.96%
	20	1	4.03%	11.88%
		1.1	10.96%	14.70%
		1.3	17.40%	28.01%
		Overall	11.23%	71.01%

Table 5.6: Average gap for the deadline problem under Case (i) with random arrivals

Percent Orders Late	Maximum Order Weight	Due Date Tightness	Average Gap	Maximum Gap
10%	10	High	1.89%	5.88%
		Medium	6.16%	16.67%
		Low	10.02%	16.67%
	15	High	1.90%	9.52%
		Medium	6.09%	11.11%
		Low	11.02%	11.76%
	20	High	2.13%	7.69%
		Medium	7.45%	13.04%
		Low	10.00%	13.64%
20%	10	High	1.89%	5.88%
		Medium	6.69%	16.67%
		Low	10.02%	16.67%
	15	High	1.90%	9.52%
		Medium	6.09%	11.11%
		Low	12.13%	16.67%
	20	High	2.13%	7.69%
		Medium	7.45%	13.04%
		Low	9.20%	13.64%
30%	10	High	1.89%	5.88%
		Medium	8.40%	16.67%
		Low	10.02%	16.67%
	15	High	1.90%	9.52%
		Medium	6.09%	11.11%
		Low	9.91%	11.76%
	20	High	2.13%	7.69%
		Medium	6.05%	13.04%
		Low	9.31%	13.64%
		Overall	6.29%	16.67%

Chapter 6

Conclusions

In this thesis, we have considered various production-distribution scheduling problems in a supply chain setting. In the second chapter, we analyzed four problems related to order assignment and scheduling in a supply chain with one or more suppliers and one customer. Computational complexity of various cases of the problems have been clarified, and polynomial-time exact algorithms have been proposed for some special cases of the problems. All the four problems are in general NP-hard, and fast heuristics have been proposed for each of them. We have analyzed the worst-case and asymptotic performance of two of the heuristics. We have also tested each heuristic computationally.

In the third chapter, we studied a due-date based problem involving one supplier and one or more customers. We saw that for an arbitrary number of customers, the problem is NP-hard even in the special case where the processing times and the due dates are agreeable. A fast heuristic has been proposed that is asymptotically optimal when the number of orders goes to infinity. Computational tests show that the heuristic is capable of generating near optimal solutions.

In the fourth chapter, we studied a joint cyclic production and distribution scheduling problem in a two-stage supply chain with one or more suppliers, one warehouse, and one customer. We have given either optimal approaches or heuristic methods to solve the problem under two policies on production and delivery cycles. For the case with common production and delivery cycle at each supplier (policy (i)), we have proved that there exists an optimal solution where the delivery cycle time from a supplier to the warehouse is an integer multiple of the delivery cycle time from the warehouse to the customer. Based on this property, we have shown that there is a closed-form optimal solution to the problem with a single supplier under policy (i), and developed an efficient heuristic for the general problem under policy (i). The problem under policy (ii), which is more general than policy (i), is solved by a heuristic approach. Both the heuristics were shown to perform well in the computational tests.

In the fifth chapter, we looked at a problem with one supplier and one customer where different jobs may have different weights for delivery. The objective was to arrive at an integrated production and distribution schedule that minimizes the total distribution cost. We minimize the number of shipments subject to time based performance measures such as deadlines or the average lead times. We considered six different cases, three for the deadline version and three for the lead time version. Except for one case for which an optimal polynomial time algorithm was given, all the remaining were shown to be NP-hard. We have given heuristics with known worst-case performance for all the NP-hard cases. We carried out computational tests to analyze the performance of these heuristcs, where the output from the heuristic was compared with a lower bound obtained through column generation. The heuristics were able to obtain optimal or near-optimal solutions for most of the problem instances tested.

As mentioned in Chapter 1, the aim of this thesis is three-fold: (i) To propose various integrated production-distribution scheduling models that closely mirror pracitical supply chain operations in some environments. (ii) To develop computationally effective optimization based solution algorithms to solve these models. Our solution approaches can be used as decision tools by practitioners. (iii) To provide managerial insights into the potential benefits of coordination between production and distribution operations in a supply chain. Many supply chains may involve more than just one or two stages. Though we have not given closed form equations or algorithms for more complicated supply chains, the insights from the study can still be carried over. In many cases, a simple coordination mechanism between two adjacent stages in a supply chain may itself prove very powerful.

The various models studied in this thesis also give an insight into the system characteristics that would benefit significantly from coordination. If a system is such that service based performance measure is not crucial, then it may not be worth the effort to coordinate the production and distribution activities. For example, in Model 1, if the lead time performance is not crucial (the value of α is low or the lead time constraint is not very binding), then the production sequence does not matter significantly. It may be sufficient to just assign the orders to plants in a cost-effective way and then deliver them in as few delivery batches as possible. Similarly in Model 2, if the deadlines are not very tight, then the sequential approaches may not differ significantly from the integrated approach in terms of performance. But in cases where the system is congested, or when the system is such that the production constraints are tight, integration will become important. Similarly, in Model 3, integration becomes more crucial as the benefits from consolidation begin to increase, for example, when the distribution activities have low fixed costs but high variable costs, or when the number of suppliers is high. So depending on the supply-chain, integrated production-distribution scheduling may result in significant savings.

Various extensions to this study could be considered. We have not explored routing options in models that involve more than one supplier or more than one customer. In all the models studied, it has been assumed that direct shipments take place between the supplier and the customer. It will be interesting to analyze the effects of introducing routing decisions into the model. We have considered a setup where the overall supply chain cost is minimized. This assumes a cooperative setup. In many instances, different parts of the supply chain may be owned by different firms. A competitive setup may be more appropriate under such circumstances. In our study, we assume that all the orders (or demand rate) are known at the beginning. A dynamic setup where orders become known over time would be an interesting extension. We did a limited study of this extension by allowing dynamic order arrivals for Case (i) of the deadline model in Chapter 5. If dynamic arrivals are allowed, order assignment and/or scheduling will also have to be carried out dynamically and it may be necessary to partially modify an existing schedule when a new order is received. All the extensions proposed here will in general be NP-hard but it may be possible to propose efficient heuristics or solve some special cases optimally in polynomial time.

Bibliography

[1] Atkins, D. and M. Queyranne, “Lot Sizing Policies for Finite Production Rate Assembly Systems”, Operations Research, 40 (1992), 126 - 141.

[2] Bagchi, P.K. and F.W. Davis, “Some Insights into Inbound Freight Consolidation”, International Journal of Physical Distribution and Materials Management, 18(6) (1988) 27-33.

[3] Bazaraa, M.S., H.D. Sherali, and C.M. Shetty, “Nonlinear Programming: Theory and Algorithms”, Second Edition, John Wiley & Sons, Inc. New York, (1993).

[4] Belenguer, J.M., M.C. Martinez, and E. Mota, “A lower bound for the split delivery vehicle routing problem”, Operations Research, 48 (2000), 801-810.

[5] Benjamin, J., “An analysis of inventory and transportation costs in a constrained network”, Transportation Science, 23(3) (1989), 177-183.

[6] Bertazzi, L. and M.G. Speranza, “Minimizing Logistics Costs in Multistage Supply Chains”, Naval Research Logistics, 46 (1999), 399-417.

[7] Blackburn, J.D. and R.A. Millen, “Improved heuristics for multi-stage requirement planning systems”, Management Science, 28 (1982), 44 - 56.

[8] Blumenfeld, D.E., L.D. Burns and C.F. Daganzo, “Synchronizing Production and Transportation Schedules”, Transportation Research, 25B (1991), 23-37.

[9] Blumenfeld, D.E., L.D. Burns, J.D. Diltz, and C.F. Daganzo, "Analyzing Tradeoffs between Transportation, Inventory and Production Costs on Freight Networks", Transportation Research, 19B (1985), 361-380.

[10] Boyaci, T. and G. Gallego, "Serial Production/Distribution Systems under Service Constraints", Manufacturing & Service Operations Management, 3 (2001), 43-50.

[11] Bramel, J., S. Goyal, and P. Zipkin, "Coordination of Production/Distribution Networks with Unbalanced Leadtimes", Operations Research, 48 (2000), 570-577.

[12] Burns, L.D., R.W. Hall, D.E. Blumenfeld, and C.F. Daganzo, "Distribution Strategies that Minimize Transportation and Inventory Costs", Operations Research, 33 (1985), 469-490.

[13] Chan, L.M.A., A. Muriel, and D. Simchi-Levi, "Supply-chain management: Integrating inventory and transportation", Working Paper, Department of Industrial Engineering and Management Sciences, Northwestern University, (1997).

[14] Chandra, P. and M.L. Fisher, "Coordination of production and distribution planning", European Journal of Operational Research, 72 (1994), 503-517.

[15] Chang, Y.-C., and C.-Y. Lee, "Machine scheduling with job delivery coordination", European Journal of Operational Research, 158 (2004), 470-487.

[16] Chen, Z.-L, "Scheduling and common due date assignment with earliness-tardiness penalties and batch delivery costs", European Journal of Operational Research, 93 (1996), 49-60.

[17] Chen, Z.-L., "Integrated Production and Distribution Operations: Taxonomy, Models, and Review", In Handbook of Quantitative Supply Chain Analysis: Modeling in

the E-Business Era, Eds: D. Simchi-Levi, S.D. Wu and Z.-J. Shen. Kluwer Academic Publishers, (2004).

[18] Chen, Z.-L. and G.L. Vairaktarakis, “Integrated scheduling of production and distribution operations”, Management Science, Forthcoming (2005).

[19] Cheng, T.C.E., V.S. Gordon, and M.Y. Kovalyov, “Single machine scheduling with batch deliveries”, European Journal of Operational Research, 94 (1996), 277-283.

[20] Coffman, Jr., E.G., M.R. Garey, and D.S. Johnson, “Approximation algorithms for bin packing: a survey.” In: D.S. Hochbaum (editor), Approximation Algorithms for NP-hard Problems, PWS Publishing, Boston, (1997), 46-93.

[21] Cohen, M.A. and H.L. Lee, “Strategic analysis of integrated production-distribution systems: models and methods”, Operations Research, 36 (1988), 212-228.

[22] Cooper, M., “Cost and Delivery Time Implications of Freight Consolidation and Warehousing Strategies”, International Journal of Physical Distribution and Materials Management, 14(6) (1984), 47 - 67.

[23] Crowston, W.B., M. Wagner, and J.F. Williams, “Economic Lot Size Determination in Multi-stage Assembly System”, Management Science, 19 (1973), 517 - 527.

[24] Dasci, A. and V. Verter, “A Continuous Model for Production-Distribution System Design”,

[25] European Journal of Operational Research, 129 (2001), 287-298.

[26] Dogan, K. and M. Goetschalckx, “A Primal Decomposition Method for the Integrated Design of Multi-Period Production-Distribution Systems”, IIE Transactions, 31 (1999), 1027 - 1036.

[27] Dornier, P.-P., R. Ernst, M. Fender, and P. Kouvelis, "Global Operations and Logistics: Text and Cases", John Wiley & Sons, Inc, (1998).

[28] Dror, M., and P. Trudeau, "Savings by split delivery routing", Transportation Science, 23 (1989), 23-37.

[29] Fumero, F. and C. Vercellis, "Synchronized development of production, inventory, and distribution schedules", Transportation Science, 33(1999), 330-340.

[30] Garey, M.R. and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H. Freeman, San Francisco, (1979).

[31] Ghosh, J.B. and Gupta, J.N.D., "Batch scheduling to minimize maximum lateness", Operations Research Letters, 21 (1997), 77-80.

[32] Hahm, J. and C.A. Yano, "The Economic Lot and Delivery Scheduling Problem: The Single Item Case", International Journal of Production Economics, 28 (1992), 235-252.

[33] Hahm, J. and C.A. Yano, "Economic Lot and Delivery Scheduling Problem: The Common Cycle Case", IIE Transactions, 27 (1995a), 113-125.

[34] Hahm, J. and C.A. Yano, "Economic Lot and Delivery Scheduling Problem: The Nested Schedule Case", IIE Transactions, 27 (1995b), 126-139.

[35] Hahm, J. and C.A. Yano, "Economic Lot and Delivery Scheduling Problem: Powers of Two Policies", Transportation Science, 29 (1995c), 222-241.

[36] Hall, L.A. and D.B. Shmoys, "Jackson's rule for single-machine scheduling: making a good heuristic better", Mathematics of Operations Research, 17 (1992), 22-35.

[37] Hall, N.G., M. Lesaoana, and C.N. Potts, "Scheduling with fixed delivery dates", Operations Research, 49 (2001), 134-144.

[38] Hall, N.G. and C.N. Potts, "Supply chain scheduling: Batching and delivery", Operations Research, 51 (2003), 566-584.

[39] Hall, R.W., "On the Integration of Production and Distribution: Economic Order and Production Quantity Implications", Transportation Research, 30B (1996), 387-403.

[40] Hammond, J.H. and A. Raman, "Sport Obermeyer, Ltd", Harvard Business School Case #9-695-022 (1996).

[41] Hariri, A.M.A and Potts, C.N., "Single machine scheduling with batch set-up times to minimize maximum lateness", Annals of Operations Research, 70 (1997), 75-92.

[42] Herrmann, J. and C.-Y. Lee, "On scheduling to minimize earliness-tardiness and batch delivery costs with a common due date", European Journal of Operational Research, 70 (1993), 272-288.

[43] Horn, W.A, "Minimizing average flow time with parallel machines", Operations Research, 21 (1973), 846-847.

[44] Jackson, J.R., "Scheduling a production line to minimize maximum tardiness", Research Report 43, Management Science Research Project, University of California, Los Angeles, (1995).

[45] Jayaraman, V. and H. Pirkul, "Planning and Coordination of Production and Distribution Facilities for Multiple Commodities", European Journal of Operational Research, 133 (2001), 394-408.

[46] Jensen, M.T. and M. Khouja, "An Optimal Polynomial Time Algorithm for the Common Cycle Economic Lot and Delivery Scheduling Problem", European Journal of Operational Research, 156 (2004), 305-311.

[47] Johnson, M.E., "Learning from toys: Lessons in managing supply chain risk from the toy industry", California Management Review, 43 (2001), 106-124.

[48] Kaminsky, P. and D. Simchi-Levi, "Production and distribution lot sizing in a two stage supply chain". IIE Transactions, 35 (2003), 1065-1074.

[49] Lee, C.-Y. and Z.-L. Chen, "Machine scheduling with transportation considerations" Journal of Scheduling, 4 (2001), 3-24.

[50] Lee, C.-Y. and Li, C.-L., "On the fixed interval due-date scheduling problem", Discrete Applied Mathematics, 68 (1996), 101 - 117.

[51] Moily, J.P. "Optimal and Heuristic Procedures for Component Lot-splitting in Multistage Manufacturing Systems", Management Science, 32 (1986), 113 - 125.

[52] Moily, J.P. and J.P. Matthews, "Procedures for Determining Relative Frequencies of Production/Order in Multistage Assembly Systems", Decision Sciences, 18 (1987), 279 - 291.

[53] Muckstadt, J. and R. Roundy, "Analysis of Multistage Production Systems", In Logistics of Production and Inventory, Handbooks in OR/MS, Volume 4, S. Graves, A. Rinnooy Kan, and P. Zipkin (eds.), Elsevier, North-Holland, Amsterdam, 1993.

[54] Owen, S.H. and M.S. Daskin, "Strategic Facility Location: A Review", European Journal of Operational Research, 111 (1998), 423-447.

[55] Pinedo, M., “Scheduling: Theory, Algorithms, and Systems, 2nd Edition”, Prentice Hall, Upper Saddle River, New Jersey, (2002).

[56] Potts, C.N., “Analysis of a heuristic for one machine sequencing with release dates and delivery times”, Operations Research, 28 (1980), 1436-1441.

[57] Potts, C.N., “Analysis of a linear programming heuristic for scheduling unrelated parallel machines”, Discrete Applied Mathematics, 10 (1985), 155-164.

[58] Potts, C.N., and Kovalyov, M. Y., “Scheduling with batching: A review”, European Journal of Operational Research, 120 (2000), 228-249.

[59] Pyke, D.F. and M.A. Cohen, “Multiproduct Integrated Production-Distribution Systems”, European Journal of Operational Research, 74 (1994), 18-49.

[60] Sabri, E. and B.M. Beamon, “A Multi-Objective Approach to Simultaneous Strategic and Operational Planning in Supply Chain Design”, Omega, 28 (2000), 581 - 598.

[61] Sarmiento, A.M. and R. Nagi, “A review of integrated analysis of production-distribution systems”, IIE Transactions, 31 (1999), 1061-1074.

[62] Schwarz, L.B. and L. Schrage, “Optimal and System Myopic Policies for Multi-echelon Production-inventory Assembly Systems”, Management Science, 21 (1975), 1285 - 1294.

[63] Shen, Z.-J. M., C. Coullard, and M.S. Daskin, “A Joint Location-Inventory Model”, Transportation Science, 37 (2003), 40 - 55.

[64] Speranza, M.G. and W. Ukovich, “Minimizing Transportation and Inventory Costs for Several Products on a Single Link”, Operations Research, 42 (1994), 879-894.

[65] Vidal, C.J. and M. Goetschalckx, "Strategic Production-Distribution Models: A Critical Review with Emphasis on Global Supply Chain Models", European Journal of Operational Research, 98 (1997), 1-18.

[66] Wang, G. and T.C.E. Cheng, "Parallel machine scheduling with batch delivery costs", International Journal of Production Economics, 68 (2000), 177-183.

[67] Webster, S. and K.R. Baker, "Scheduling groups of jobs on a single machine", Operations Research, 43 (1995), 692 - 703.

[68] Williams, J.F., "A Hybrid Algorithm for Simultaneous Scheduling of Production and Distribution in Multi-Echelon Structures", Management Science, 29 (1983), 77 - 92.

[69] Woeginger, G.J., "Heuristics for parallel machine scheduling with delivery times", Acta Informatica, 31 (1994), 503-512.

[70] Woeginger, G.J., "A polynomial-time approximation scheme for single-machine sequencing with delivery times and sequence-independent batch set-up times", Journal of Scheduling, 1 (1998), 79-87.

[71] Yuan, J., "A Note on the Complexity of Single-Machine Scheduling with a Common Due Date, Earliness-Tardiness, and Batch Delivery Costs", European Journal of Operational Research, 94 (1996), 203-205.